

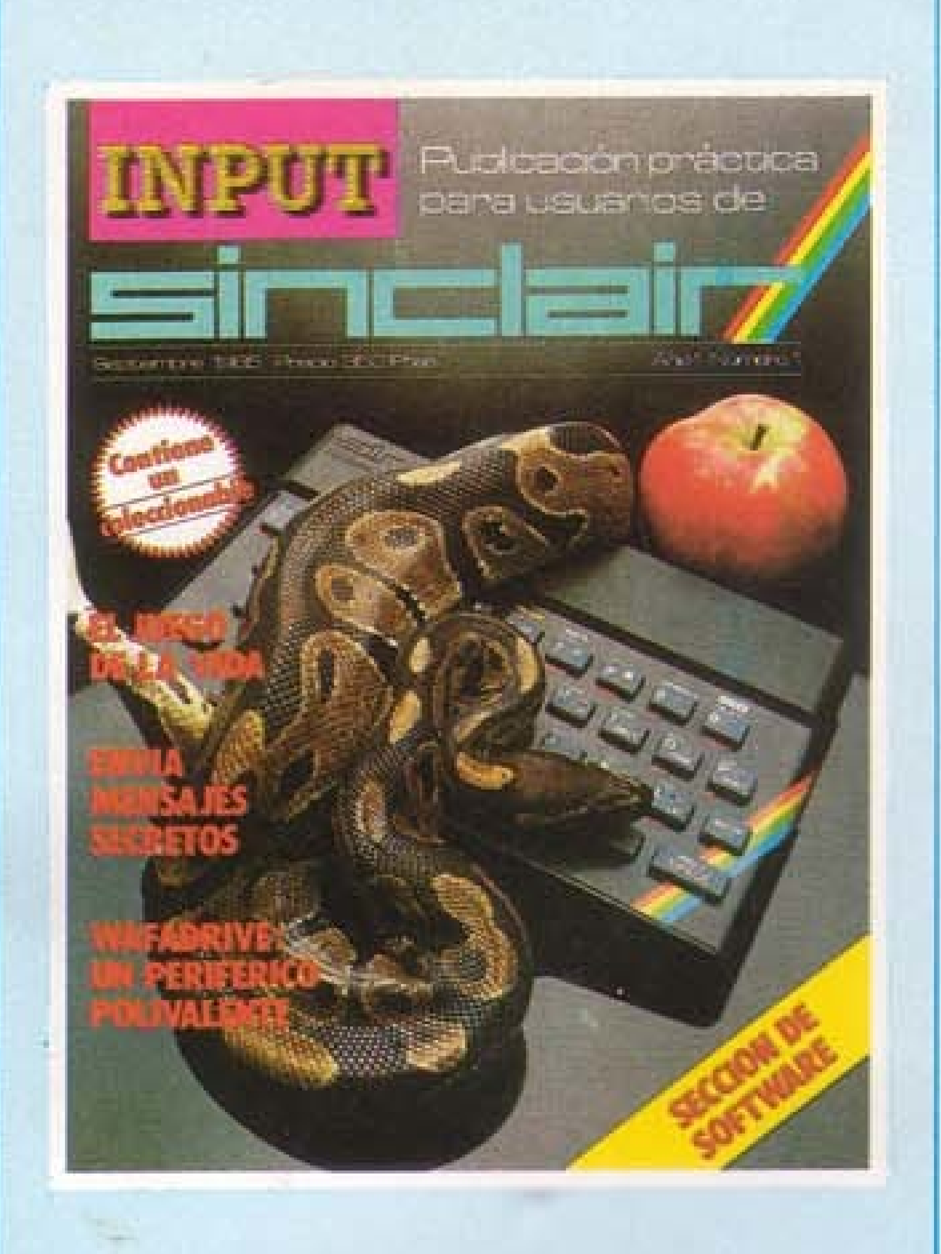
INPUT
Publicación práctica para usuarios de
sinclair
Septiembre 1985 Precio 350 Ptas
Año 1 Número 1
Contiene un coleccionable
EL JUEGO DE LA VIDA
ENVIA MENSAJES SECRETOS
WAFADRIVE: UN PERIFERICO POLIVALENTE
SECCION DE SOFTWARE


STARS
SPECTRUM
El SOFTWARE a la altura de tu ordenador.
Nº0
CADA MES EN TU KIOSKO



Sumario

AÑO 1 NUMERO 1



DIRECTOR:
Alejandro Diges
COORDINADOR EDITORIAL:
Francisco de Molina
DISEÑO GRAFICO:
Tomás López
COLABORADORES:
Antonio Taratiel, Luis R. Palencia, Christophe Pais, Francisco Tórtola, José Pérez Montero, Benito Román
INPUT Sinclair es una publicación juvenil de EDICIONES FORUM

GERENTE DIVISION DE REVISTAS:
Angel Sabat

PUBLICIDAD: Grupo Jota
Madrid: c/ Cuenca, 1, 1.º
Tel. 253 45 01 /02
Barcelona: Avda. de Sarriá, 11-13, 1.º
Tel. 250 23 99
FOTOMECANICA: Ochoa, S.A.
COMPOSICION: Fernández Ciudad, S.L.
IMPRESION: Edime, S.A.
Depósito legal: M. 27.885-1985
Suscripciones: EDISA,
López de Hoyos, 141. 28002 Madrid
Tel. (91) 415 97 12
Redacción:
Alberto Alcocer, 46, 4.º
28016 Madrid. Tel. 250 10 00
DISTRIBUIDORA:
R.B.A. PROMOTORA DE EDICIONES, S. A.
Travesera de Gracia, 56. Edificio Odiseus.
08006 Barcelona
Se solicitará el control OJD

YA ESTAMOS AQUI

¡Una revista más!, será la reflexión que haréis muchos lctores. Efectivamente así es, salimos a la calle para ampliar la oferta disponible en los kioscos. Sin embargo, nacemos con pretensiones, resumidas en un objetivo único: ser el centro de vuestras preferencias. Intentaremos lograrlo número a número.

Hemos creado diversas secciones que suponemos serán del interés de la mayoría de vosotros. Conviene hacer especial mención de la titulada Educación. En los primeros números se describe el **Logo** y su modo de empleo, pero trataremos otros temas relacionados con dicha temática.

En las páginas centrales se incluye una sección que enseña a programar juegos. El grado de complejidad irá aumentando, tratándose diversos tipos, desde los más sencillos a los más complejos (juegos de guerra e inteligencia), partiendo desde el comienzo de lo más básico. Esta sección puede ser coleccionada por quienes así lo decidáis.

El *software* alcanza una especial relevancia en **Revista de Software,** sección en la que, mes tras mes, haremos repaso de las novedades más interesantes. Los programas se describen con detalle, comentando diversos trucos y tácticas. Cada programa tiene su ficha técica y una calificación de sus diversos aspectos. Y, cómo no, se incluyen numerosas fotografías de pantallas.

A lo largo de la revista podréis encontrar diversos concursos. Desde luego, no perseguimos captar a coleccionistas de regalos, pero sí corresponder a vuestra participación.

Confiamos en que **INPUT** sea de vuestro agrado, pero os invitamos a que nos escribáis con vuestras sugerencias.

EL BUZON DE INPUT

El próximo número de INPUT reservará un espacio para responder a vuestras cartas.

No pretendemos que sea un rincón únicamente destinado a la resolución de dudas y problemas. Por supuesto que intentaremos proporcionaros este tipo de ayuda, pero nuestra pretensión es tener algo más: un espacio para la comunicación donde nos contéis a todos lo que hacéis con el ordenador, proyectos, intenciones, cómo es vuestra instalación y las cien mil cosas que se os ocurran.

Dirigir las cartas a:

EL BUZON DE INPUT SINCLAIR
Alberto Alcocer, 46, 4.° B
28016 Madrid

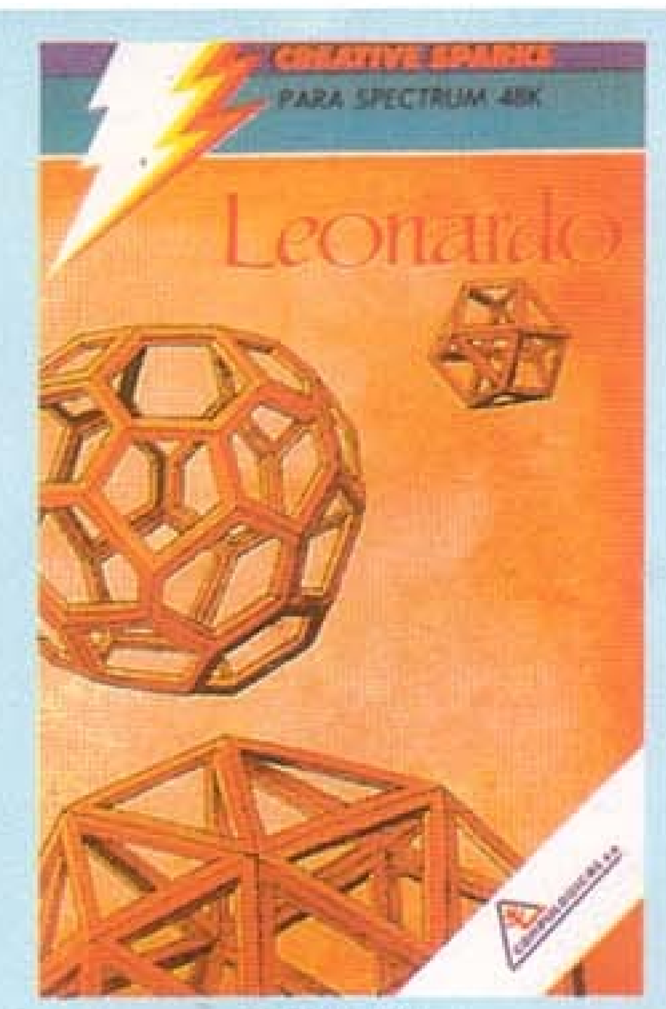
MEJORAS Y APLICACIONES

Cualquier programa publicado en INPUT puede sugerir nuevas ideas. Te lanzamos una propuesta a modo de reto: mejora y/o encuentra aplicaciones a nuestros listados. No olvides que muchas veces salta una chispa que trae una solución genial.

Cada tercer mes, a partir de la publicación de un número, elegiremos la mejor aportación.

La decisión será inapelable, pero tener por seguro que será ecuánime y, sobre todo, resuelta por un grupo de expertos.

El ganador recibirá un premio consistente en una suscripción por un año a INPUT SINCLAIR.



SPECTRUM

Un paquete gráfico de fácil manejo que puede ser utilizado por artistas amateurs, delineantes o programadores que deseen realizar sus programas con estupendos gráficos y sofisticadas técnicas gráficas.

Leonardo contiene más comandos que cualquiera de sus rivales. El programa incluye una completa colección de ejemplos de cómo obtener los mejores resultados de más de 40 comandos.



Santa Cruz de Marcenado, 31 - 28015 Madrid - Teléf. 241 10 63

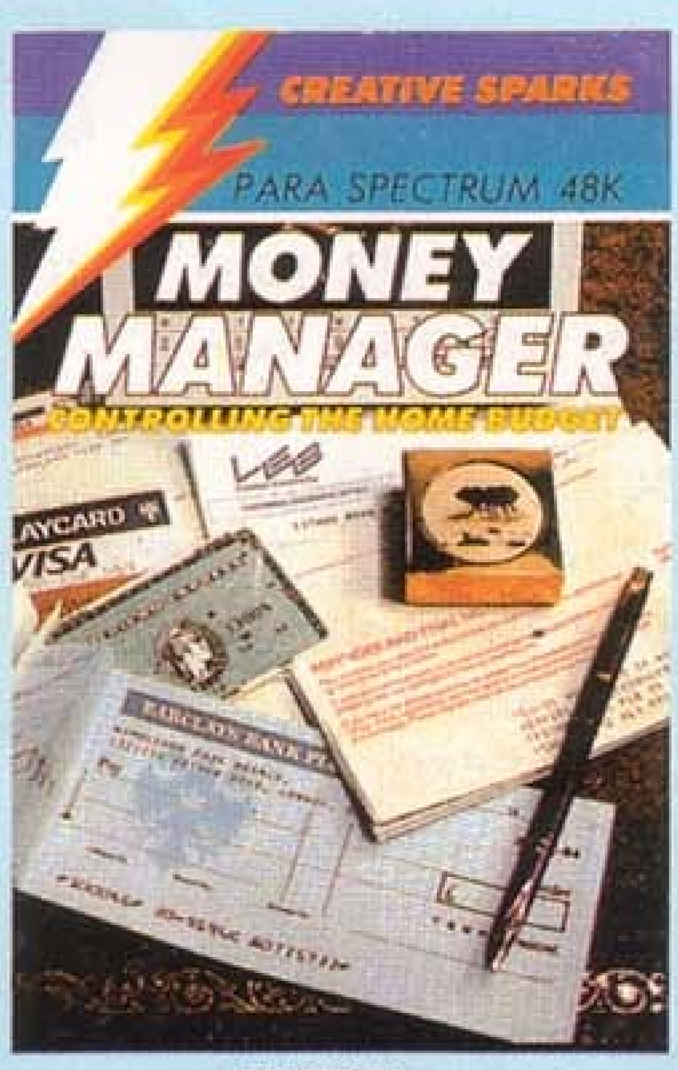
MIND GAMES
AMERICAN FOOTBALL



C-64 AMSTRAD SPECTRUM

Experimente la trepidante acción del FUTBOL AMERICANO.
- Para 1 ó 2 jugadores.
- Rápidos gráficos animados.

Solicite nuestro amplio catálogo

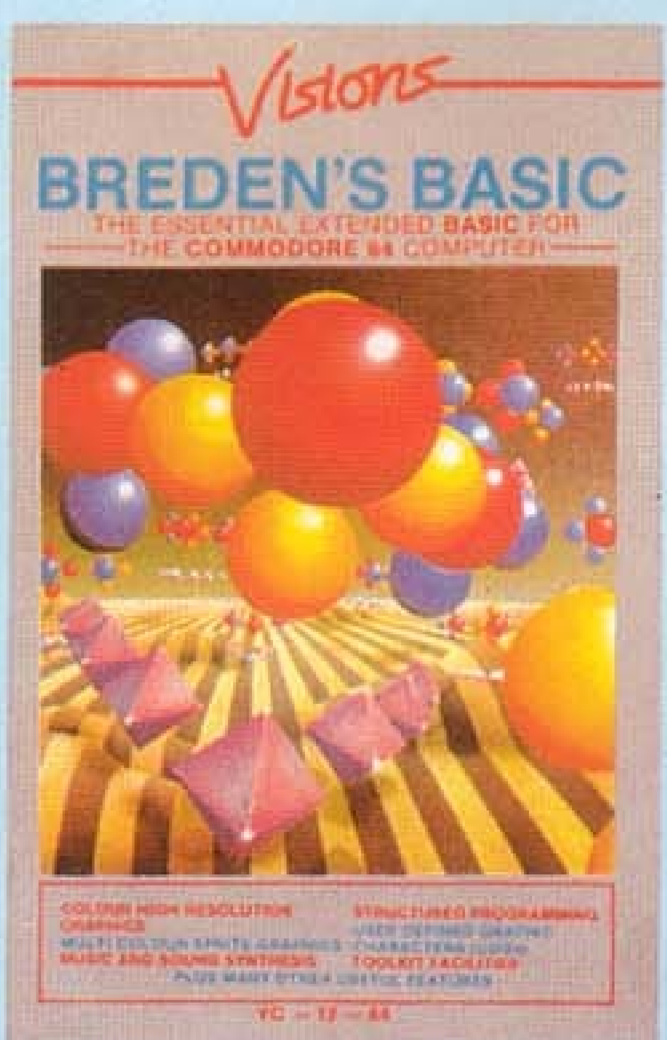


SPECTRUM

Todo el mundo de las finanzas a su alcance por el precio de un juego.

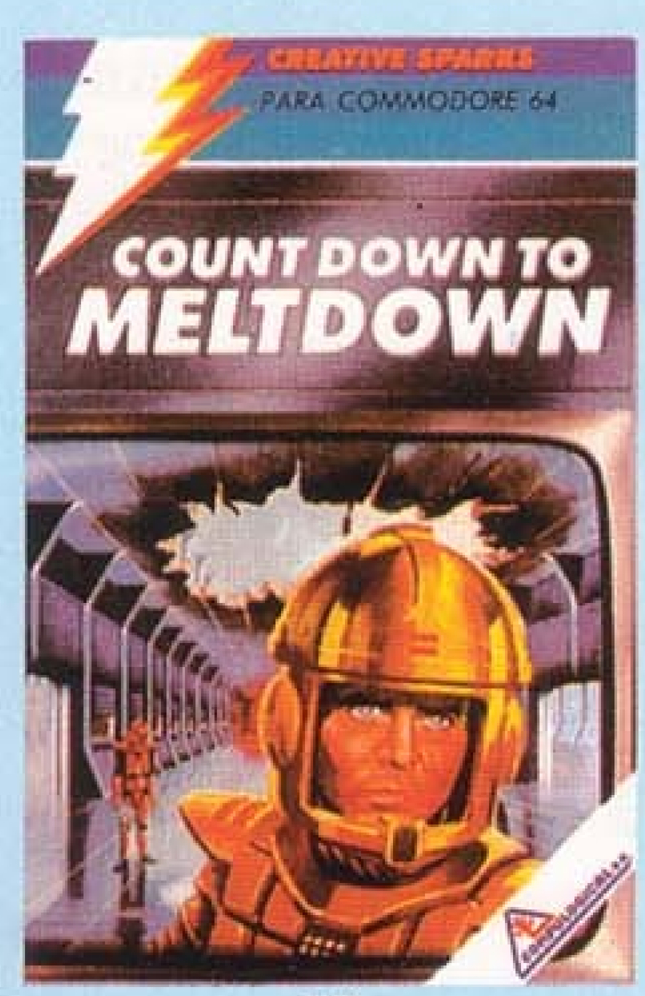
Se puede guardar toda la información en una cinta cassette.

Cualquier información de pantalla se puede sacar impresa con una impresora ZX.



C-64

La extensión esencial del BASIC para su COMMODORE-64. BREDEN'S BASIC, programa escrito en código máquina aporta 135 nuevos comandos y permite obtener los mejores efectos en sonido y gráficos.
Estos comandos abarcan los siguientes conceptos:
- Manejo de sprites.
- Control de gráficos y pantalla.
- Música y sonido.
- I/O (entrada y salida).
- Disco.
- Manipulación numérica.
- Detección de errores.
- Ayuda al programador.
- Programación estructurada.



C-64

Un divertido juego de aventuras y acción que le pone al mando de un equipo de androides intentando alcanzar el centro de un reactor nuclear antes de que explote. Este juego tiene cerca de 2.000 habitaciones para explorar, y cada una de ellas es de 3 dimensiones.

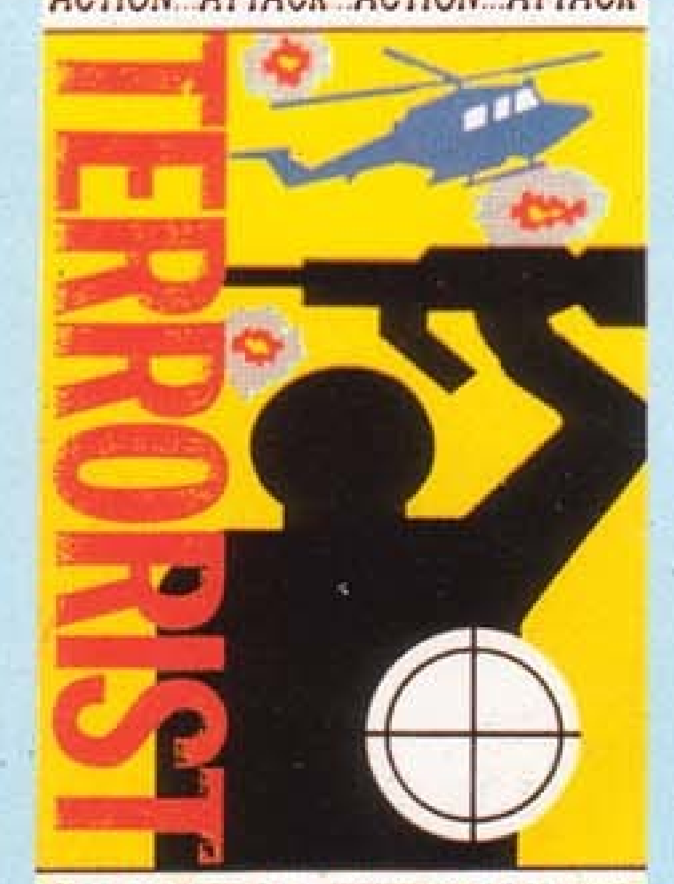
Probablemente el juego más excitante hasta ahora desarrollado para el C-64.

- Rápida carga en cassette.



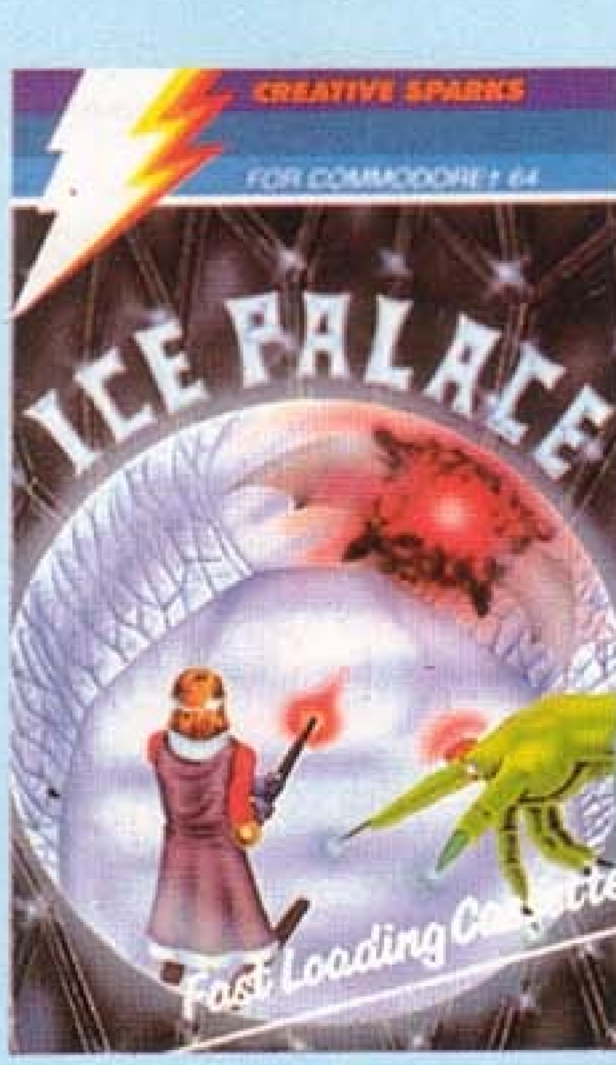
SPECTRUM

- 240 lugares especialmente detallados, muchos de los cuales contienen sorpresas ocultas y los ingredientes necesarios para completar el juego.
- Robots interactivos.
- Una tremenda aventura de arcada con abundancia de increíbles gráficos.



C-64

Defienda su ciudad contra los terroristas!
- Divertidas secuencias de batallas en 3D.
- Elección de estrategias múltiples.
- Secuencias de ataque constantemente cambiantes.



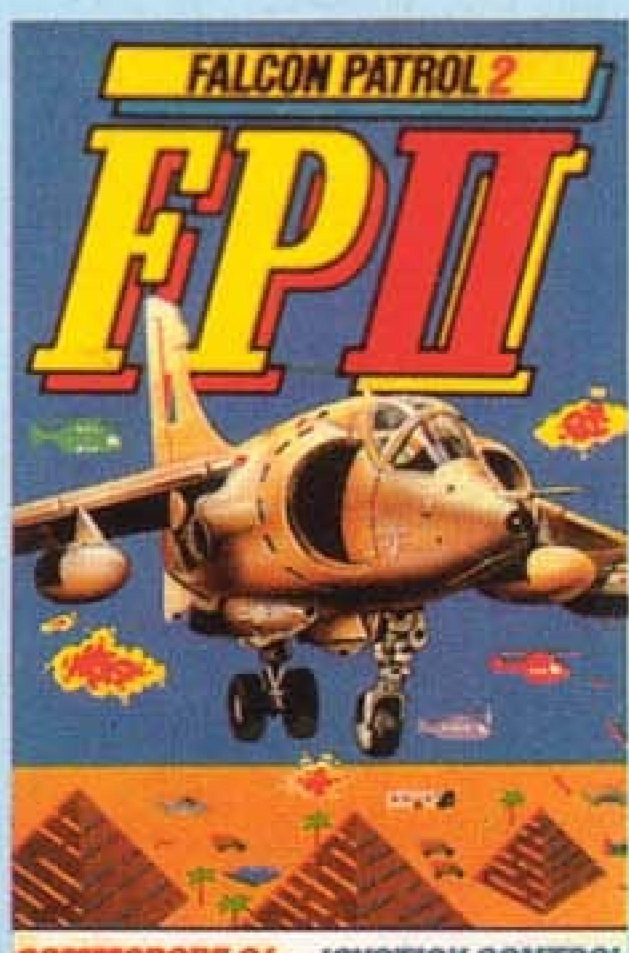
C-64

Salve a la Tierra del Terrible Hechizo del Hielo!
- Acción en tiempo real.
- 1.200 lugares repartidos en 7 niveles de juego.
- Control joystick.



C-64 AMSTRAD SPECTRUM

Atrévase a penetrar en el mundo de las tinieblas!
- Gráficos en 3D magnificamente detallados.
- 17 pantallas para realizar.
- Emocionante test de su coordinación y estrategia en este juego que provoca adicción.



C-64

- Increíbles gráficos en 3 D.
- 16 niveles de juego con dificultad en aumento.
- Interferencia de radar.
- Helicópteros ametralladores.
- Misiles tierra-aire y aire-aire.



¿ JACK TRAMIEL EN ESPANA?

Se dice, se comenta, se afirma que Jack Tramiel, fundador y expresidente de Commodore, visitó Espana durante los primeros días del pasado mes de julio.
Tramiel se situó a la cabeza de Atari, después de los problemas que le condujeron a dejar la presidencia de Commodore en los primeros meses del 85.
Al parecer su visita tendría como objetivo principal entrevistarse con un importante distribuidor nacional.Las especulaciones van más lejos de suponer que se pretende una implantacion más fuerte de Atari en nuestro país, incluso llega a comentarse la posibilidad de fabricación, aun cuando Tramiel tiene resuelto el problema en Taiwan, donde se encuentra mano de obra a bajo precio.
Los nuevos modelos de Atari compiten en los mismos mercados que Commodore y Sinclair.
Unos días antes, a mediados de junio, Tramiel llegó a afirmar en Francia que se hallaba en conversaciones con el gobierno de aquel país de cara a participar en un plan de fabricación de cien mil ordenadores en colaboración con un fabricante francés, de quien no especificó el nombre.
Igualmente expreso su idea de crear dos nuevas fábricas, en Europa y los EE.UU.
Habrá que esperar acontecimientos. Tal vez el cercano SIMO traiga noticias interesantes para la Distribución.



TAL VEZ SEA PORTATIL

Sinclair Research ha confirmado recientemente los rumores que hace tiempo venian escuchandose insistentemente.
Según esto, sus laboratorios de investigación y desarrollo están trabajando en el prototipo de un nuevo ordenador doméstico.
Se supone que el nuevo modelo estará basado en una combinación del Spectrum y la pantalla plana de su televisor.
Igualmente salió a la luz un interesante proyecto, que representará un cambio radical en la forma de almacenamiento masivo de datos. Su desarrollo tambien está en marcha de cara al QL.
Será un dispositivo de estado sólido (sin piezas móviles), cuyo corazón está ocupado por una oblea de silicio, que contendrá una memoria RAM capaz de guardar hasta medio Mbyte de información. Para que la informacion no se volatilice, el conjunto irá provisto de una bateria propia, que alimentará constamente al dispositivo.



MAS REVISTAS MAGNETICAS

Este no es el único producto de la linea puesto en la calle por Ventamatic. De su cabecera Spectrumania va por el tercer número, incorporando titulos ineditos y otros previamente conocidos, tales como Autopista Galactica, Fruit Machine. Tambien hay presentes algunas utilidades.
La inquieta empresa se introduce ahora en los sistemas musicales controlados por ordenador, eligiendo el estandar MIDI como centro de sus actividades.
Ventamatic, firma comercializadora de microinformática, lanza durante los próximos dias una nueva revista en soporte magnético para el Spectrum, en la que incluye juegos tales como el World Destruction o una Contabilidad Personal, que puede trabajar con 20 cuentas y 1.400 asientos, generando hasta 64 caracteres por linea.
El p.v.p. previsto es de 695 pts y lleva por título *Spectrum Top Hits*.



¿ NUEVO QL?

Cada vez parece mas probable que Sinclair lance una nueva versión del QL, que dispondría de nuevo *software* incluido de forma interna en *chips* de ROM. Una opinión distinta afirma que se trataría de cartuchos externos.
El *software* es de la firma Psion y llevaría cuatro paquetes integrados, los mismos que emplea la versión del QL utilizada por el sistema de Automatización de Oficinas, One per Desk.



HOLOGRAMA ANTI-PIRATAS

Preocupadas por el caríz que viene tomando el asunto de la copia ilegal de programas, las empresas estan buscando métodos cada vez mas seguros.
Una solución novedosa, que no afecta al contenido de la cinta, es la aparición de etiquetas holográficas, cuyo aspecto es el de un trozo de papel metalizado.
La difícil, por no decir imposible, reproducción exacta asegura el origen de (al menos) la caratula impresa.



EL COCHECITO

El esperado éxito del vehiculo de tracción electrica de Sinclair, bautizado por algunos como "El Clivemovil" no ha sido todo lo afortunado que esperaba su promotor.
Las ventas fueron tan reducidas, que en los momentos de mayor tensión, provocada por la falta de liquidez financiera que dejó a *sir* Clive sin la presidencia de la compañia, se consideró incluso la posibilidad de vender Sinclair Vehicles, la firma creada para fabricar el automovil.
De todas formas, la producción del C5 se vió reducida de un modo drástico, las mil unidades semanales, que salian de fábrica en un principio, se vieron reducidas a solo cien y de los cien trabajadores que se ocupaban del montaje, noventa fueron reciclados para seguir montando lavadoras. Recordemos que el motor empleado por el C5 pertenece originalmente a uno de dichos electrodomésticos.
Los rumores, siempre lo mismo, afirman que el vapuleado genio no se da por vencido y está pensando en un modelo C6.
Las razones del fracaso parecen apuntar hacia la falta de seguridad que tiene el vehiculo para con su ocupante. Su altura es demasido baja.



LAS SUPEROBLEAS DE SINCLAIR

Los pasados problemas que afectaron profundamente a la estabilidad financiera de Sinclair Research no parecen haber hecho mella en la moral de *sir* Clive.
Parece bien cierto que no abandona la idea de montar una planta para fabricar revolucionarios *chips* semiconductores. La inversión necesaria se aproxima a los cincuenta millones de Libras (multipliquen!) y esta buscando a quien los aporte.
Lo revolucionario del proyecto es la tecnología de fabricación que pretende emplear. Normalmente, un ordenador es desarrollado y fabricado en base a una serie de de *chips* mas o menos estándar. Por ejemplo, pocos son los sistemas que emplean un microprocesador exclusivo y tres cuartos de lo mismo ocurre con la RAM.
En el proceso de fabricación se produce simultaneamente una gran cantidad de *chips* sobre una oblea circular de silicio, que posteriormente se trocea, comprobandose cuales son los finalmente útiles.
La intención de Sinclair es contener el equivalente a varios *chips* distintos en una sola oblea. De este modo se abarata la fabricación, eliminado pasos del proceso, y se consiguen mayores velocidades de funcionamiento, por disminuirse la longitud de las pistas de conexión, que pasan de estár en la tarjeta de circuito impreso a formar parte de la oblea.
Hace algo mas de un par de años una empresa americana, llamada Trilogy, comenzó a trabajar en el mismo sentido, teniendo previsto lanzar el primer producto ahora, durante el 85. Sin embargo al poco tiempo de comenzar el trabajo aparecieron las dificultades, debidas principalemente a las tecnologías de fabricación. La elevada escala de integración de los componentes producía una cantidad de calor peligrosa. El proyecto fue teniendo cada vez mas problemas y en la actualidad está prácticamente en vía muerta. El hombre que consiguió que importantes firmas invirtieran en Trilogy, Gene Amdahl, tiene en cierto modo algo en común con Sinclair. Ambos montaron una gran empresa de ordenadores partiendo desde abajo y hoy la empresa no es suya.
De todas formas, Sinclair se permitió el lujo de criticar aquel proyecto, diciendo que el no pretende reducir un ordenador a un solo *chip*, sino incluir todos los *chips* en una oblea. De momento sigue buscando el dinero que le permita ver realizado su sueño.



EL TELEVISOR PLANO DE SIR CLIVE



Una idea acariciada por Sinclair desde veinte años atras. Su televisor plano es una realidad presente en las tiendas inglesas desde hace casi dos, habiendo pasado sin pena ni gloria. Es en los EE.UU. donde ha obtenido mayor éxito, siendo incluso elegido por American Express para promoción.
Es en los últimos tiempos cuando Dixon's, una importante red de distribución de electrónica de consumo, quien lo ha incorporado en un lote junto con la antigua impresora ZX, el Spectrum y varios *cassettes* de juegos. El invento no es nuevo, en la foto adjunta aparece el modelo que ya presentase en 1981, pero que actualmente tiene un aspecto mas similar a una radio de bolsillo.
La gran innovación es el tubo catódico, que lleva el cañon de electrones alineado con la superficie de la pantalla.
El televisor tiene la peculiaridad de poder ser sintonizado en cualquier pais del mundo, independientemente del estandar empleado

COPIAR, COPIAR HASTA MORIR

Adapso, una asociación americana que aglutina a los mas importantes fabricantes de *software*, ha lanzado un agresiva campaña publicitaria destinada a crear conciencia entre quienes realizan copias ilegalmente. Comienza uno de sus anuncios con cuatro lineas de texto que dicen:
"*Es fácil hacer una copia.*
Es rápido.
Es ilegal.
Es incorrecto."
Continua haciendo una reflexión no por evidente menos cierta: "*Es difícil de creer. Gente que no habría pensado en sustraer un producto de software de una tienda durante su hora de comida, no se lo piensa dos veces a la hora de regresar a la oficina y hacer varias copias ilegales de un mismo software.*"
Termina ofreciendo un panfleto, que gustosamente enviará la srta. Priscila a quienes se tomen la molestia de escribir a sus oficinas.
El problema tiene visos de gravedad, al menos en el terreno económico. Una encuesta llevada a cabo a principios del presente año revelaba que a la industria británica del *software* la pirateria le cuesta venticuatro millones de Libras esterlinas anuales. Casi nada!. Quienes parecen llevar la peor parte son los fabricantes de programas de juegos, que se sienten mas vulnerables que las firmas que centran su actividad en el software de gestión.
El informe alcanza tales grados de sutileza, que incluso se establecen categorias en la especialidad: Falsificaciones, imitaciones que parten de una idea básica ajena, copias hechas por minoristas para favorecer la venta de un equipo, copieteo aislado entre amiguetes, la copia mas o menos organizada que llevan a cabo los estudiantes y la rotación del personal que trabaja en las empresas del sector.

SINCLAIR AL ATAQUE

No corren buenos tiempos para *sir* Clive Sinclair. Hasta hace pocos meses era dueño de su emporio, Sinclair Research, sucesora de otra empresa, (Sinclair Radionics). Este genio de la técnica siempre ha perseguido poner la última tecnología electrónica al alcance de todos los bolsillos. A finales de los sesenta competía con los japoneses, poniendo a la venta una radio-llavero, después continuó con las cadenas de alta fidelidad a bajo coste, fabricó circuitos integrados, lanzó la primera calculadora europea de bajo precio y tamano-en versiones normal y científica. El primer reloj digital en *kit* (para montar uno mismo) que además empleaba teclas sensibles al tacto. Un largd etcetera puede seguir a estos ejemplos.
Finales de los setenta, comienza a vislumbrarse el *boom* de los microordenadores, pero su precio no es asequible. Este es un nuevo reto para Mr. Sinclair y en mayo del ochenta lanza el primer microordenador de muy bajo precio, el ZX80, el cual se puede comprar totalmente ensamblado o en forma de *kit*. El ZX81 fue el sucesor, un modelo mas compacto y de mejores prestaciones. Pero el Spectrum es el gran acierto. Se vende como rosquillas calientes y la demanda aumentaba por dias. Se fuerza la producción, pensando que la campaña navideña del ochenta y cuatro iba a evolucionar en la misma linea que la anterior. Sin embargo las cosas no salen como estaba previsto y Sinclair Research sufre un fuerte revés financiero. LLegan las vacas flacas y hay que buscar fuentes que produzcan liquidez inmediata. Como solución rapida a corto plazo se reducen los precios de venta, propiciandose la guerra entre marcas.
Por otra parte se lanza el Spectrum Plus, la version del modelo anterior con carcasa y teclado diferentes y, además, se mantiene el mismo precio. Los minoristas se dan cuenta que a igualdad de condiciones el cliente prefiere el Plus y se cuestionan la rentabilidad de sus *stocks* de modelos antiguos. Su decisión es casi unanime y consiste en no hacer pedidos si no son ofrecidas determinadas garantias.
A principios de mayo, en los almacenes de Sinclair Research había trescientas mil máquinas, lo que obligó a reducir la semana laboral en la fabrica a tres dias por semana, fijandose el objetivo de producir doscientas mil unidades.
Esto complica la situación de *sir* Clive, que era poseedor del ochenta y cinco por ciento de la compañia y poco antes del verano fue rescatado por Robert Maxwell, propietario del grupo editorial británico Pergamon Press, que publica el diario Daily Mirror.
Pergamon aportó doce millones de Libras, lo que le ha permitido hacerse con el setenta y cinco por ciento de la empresa. Este grupo dispone de tres diarios y trescientas cincuenta revistas técnicas y, además, edita *software* a traves de su filial Mirrorsoft

EL FUTURO DE SINCLAIR

El papel de *sir* Clive queda relegado a ser el consejero de investigación. Pero el sigue con proyectos. Entre otros se habla de una versión del Spectrum con 128 K, que habría de recurrir a la conmutación de bancos de memoria, puesto que su microprocesador puede direccionar hasta un máximo de 64 K.
Otro proyecto con mayores visos de realidad es el conocido con el nombre clave de Pandora, una versión portatil del Spectrum.
La fabricación de *superchips* y la memoria RAM permanente de 500 Kbytes para el QL son los nuevos objetivos.


KEMPSTON
MICRO ELECTRONICS LTD
NEW GRAND PRIX SERIES!
GRAHAM HILL BEND
PADDOCK HILL
HAILWOOD RISE
DRUIDS
FORMULA
Pedidos contrareembolso a Key Informática, S. A. Envíenme los siguientes artículos.
Nombre
Dirección
Código Postal
1 Kempston Joystick Interface 2.990 Ptas.
2 Centronics 'E' Interface 12.990 »
3 Formula 2 Joystick 3.990 »
4 Kempston Pro Joystick Interface 5.990 »
5 Formula 1 Joystick 5.990 »
6 Junior Pro Joystick 2.590 »
Incluyo talón nominativo por ptas.
Contrareembolso
Key Informática, S. A. Embajadores, 90 28012 MADRID
Tel. (91) 227 91 99
De venta en
COMPUTIQUE
Y en las mejores tiendas de informática
DANGER: THESE PRODUCTS MAY SERIOUSLY DAMAGE OUR COMPETITORS MARKET.
WARNING!
COMPETITOR/
MECHANIC
1985
Nº 02232
BRANDS

UN PERIFERICO POLIVALENTE

- UN INTERFACE QUE VALE POR MUCHOS
- LOS WAFERS
- COMANDOS DEL WAFADRIVE
- CONECTORES E INTERFACES

El ZX Spectrum está considerado como el importante promotor de un suculento mercado. Su evolución va relacionada con el número de unidades que hay funcionando en los hogares de media Europa. Los estantes de las tiendas están cada vez más poblados por nuevos elementos de apoyo al aprovechamiento del ordenador. Sus productores son, en gran mayoría, pequeñas empresas cuya viabilidad era impensable una docena de años atrás.



Aspecto de la parte frontal. De acuerdo con un diseño muy sencillo, su única conexión con el ***Spectrum*** *es un cable plano de unos 9 cm.*

A medida del **Spectrum** se fabrica de todo, desde *software* en distintos grados de sofisticación hasta *joysticks,* tabletas gráficas o impresoras especializadas. Sea cual sea el número de complementos que haya comprado el usuario más caprichoso, siempre le faltará alguno.

El **Wafadrive,** que hoy revisamos, es un conglomerado de periféricos y accesorios en una sola carcasa. No hemos dudado en rebautizarlo como «el periférico orquesta», a semejanza de esos músicos que acoplan varios instrumentos sobre su cuerpo.

Una simple lectura de sus características lleva a pensar que el **Wafadrive** tiene como meta principal competir con el **Interface 1** y el **Microdrive** de **Sinclair.** Hasta el momento ha sido muy poca la rivalidad creada en torno a la unidad de almacenamiento masivo diseñada para el **Spectrum** por su fabricante.

Cierto es que el **Microdrive** es seguido por una aureola de baja fiabilidad y desproporcionada relación prestaciones/precio. Pero no es menos cierto que muy pocas casas de *software* se han aventurado a lanzar programas en microcartucho.

El aspecto sólido y robusto que ofrece el **Wafadrive** contrasta con la graciosa imagen del **Microdrive.** Sin embargo, el tamaño del primero es bastante mayor, por incorporar dos unidades de almacenamiento de datos independientes entre sí.

Los programas y datos se guardan en pequeños *wafers,* o tabletas, de aspecto similar al microcartucho, pero prácticamente del doble de tamaño. Estos *wafers,* que dan nombre al dispositivo, se comercializan con tres capacidades máximas: 16 K, 48 K y 128 K. Todas ellas son menores que los 170 K de **Sinclair,** pero se logra reducir el tiempo medio de localización de un programa. Como se recordará, el microcartucho, al igual que los *wafers,* lleva en su interior una cinta magnética sin fin (el final de la cinta va unido a su principio) y los datos son grabados secuencialmente; es decir, un fichero detrás de otro, como en fila india. Por tanto, si buscamos un programa situado en una parte de la cinta distante del lugar en que se encuentra la cabeza de lectura, habrá un tiempo de espera proporcional a la cantidad de cinta que haya entre medias. Será lógico que este tiempo sea menor en la cinta de 16 K, aunque la capacidad total sea menor.

DISEÑO EXTERNO

El aspecto exterior del **Wafadrive** es sólido y robusto. Es una carcasa paralelepípeda, que no dispone de más adornos que algunas estrías y una etiqueta con las referencias del fabricante sobre la superficie de plástico negro.

En la parte frontal hay dos estrechas ranuras horizontales, ocultas tras dos tapas que se desplazan hacia el interior cuando insertamos un *wafer* por cualquiera de ellas. Son el

mi computer 1

CURSO PRACTICO DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR



TODAS LAS SEMANAS EN SU KIOSCO





*El **wafadrive** aparece como un sistema mucho más compacto que el conocido tándem **Interface-1/Microdrive.***

acceso a sendos mecanismos de arrastre de la cinta y a las cabezas de lectura/escritura magnética. Las tapas tienen por misión proteger al conjunto del polvo y las agresiones externas.

En el espacio que media entre ambos accesos aparecen tres LEDs (Diodos Emisores de Luz). Se iluminan para indicar el funcionamiento del dispositivo. Así, el central indica que llega correctamente la alimentación eléctrica procedente del **Spectrum.** Los otros dos se encienden cuando es utilizada la unidad de lectura/escritura físicamente más próxima.

Los wafers *tienen prácticamente el doble de tamaño que el microcartucho.*



Partiendo de la base de la carcasa fluye un cable plano, que termina en un conector enchufable directamente en el *slot* de expansión del **Spectrum.** Esta es la única conexión con el ordenador.

Aparte de la cara posterior no hay ninguna otra de la que podamos comentar algo. Volteando la caja vemos tres ranuras, que dejan asomar a otras tantas lengüetas sobresalientes directamente de la tarjeta principal de circuito impreso. Una serie de diminutas pistas de cobre las convierten en conectores.

De izquierda a derecha, la primera ventana es réplica exacta del *slot* de expansión del **Spectrum,** que resultaba inutilizado por la conexión del **Wafadrive.** Su existencia permite que sea posible conectar otros periféricos y accesorios simultáneamente.

Las otras dos responden a tareas de transferencia de datos. Casi en el centro tenemos un *interface* tipo **Centronics,** para establecer la comunicación en modo paralelo.

La última es un interface RS232, que trabaja en modo serie. Paralelo significa que los bytes se transfieren de una vez; es decir, los ocho bits que lo componen viajan paralelamente. Serie implica que un bit va detrás del otro.

En el **Wafadrive,** el *interface* **Centronics** (bautizado en recuerdo de un fabricante de impresoras que lo impuso) es unidireccional, o lo que es igual: los datos sólo pueden enviarse en una dirección desde el **Wafadrive** hacia, por ejemplo, una impresora. Por el contrario, el RS232 se diseñó para funcionar bidireccionalmente y, por tanto, el ordenador puede tanto enviar cómo recibir información del exterior.

El fabricante ha dispuesto igualmente la posibilidad de que el *interface* **Centronics** pueda emplearse para la conexión de *joysticks,* describiendo en el manual la técnica a utilizar. Para este fin existe un adaptador adecuado.

PUESTA EN MARCHA

La conexión del **Wafadrive** es tan simple como se describía hace unos párrafos. Con el ordenador desenchufado, se inserta el conector de enlace. A continuación se alimenta del modo habitual y, si todo es correcto, se ilumina el LED central. Sin embargo no quiere decir que podamos utilizar el dispostivo inmediatamente. Habrá que inicializar el sistema con NEW*. En este momento accedemos a una ampliación con respecto al BASIC disponible en el **Spectrum.** Los comandos se utilizan de un modo directo, pues son los mismos que vienen serigrafiados en las teclas del ordenador, aunque toman personalidad propia porque la sintaxis varía.

En realidad lo que sucede es que el **Wafadrive** impone su EBASIC. Los 8K más bajos de la ROM del **Spectrum** son desconectados con NEW* y son tomados en el mapa de memoria por la ROM de este periférico polivalente. También se hace con 2K de la RAM para otros usos.

De igual modo que ocurre con los microcartuchos de **Sinclair,** los *wafers* nuevos deben ser formateados



695,– Ptas.
para Spectrum 48K y PLUS.
NUEVO E INEDITO · GARANTIA DE CARGA
SPECTRUM TOP HITS 1
48K/PLUS
¡YA ESTA A LA VENTA! COMPRALA EN TU QUIOSCO HABITUAL
WORLD DESTRUCTION
21 pantallas de fantásticos gráficos animados. Integramente realizado en Código Máquina.
BOLETIN DE PEDIDO
Enviar a: VENTAMATIC - c/ Córcega, 89, entlo.
08029 BARCELONA Tel.: (93) 230 97 90
Deseo:
☐ Recibir el Nº 1 de SPECTRUM TOP HITS al precio de 695,– ptas.
☐ Suscribirme por 6 números de SPECTRUM TOP HITS, a partir del Nº ______ al precio de 4.000,– ptas.
☐ Recibir el CATALOGO COMPLETO VENTAMATIC (32 páginas) de artículos de microinformática, al precio de 200,– ptas., a deducir de mi próximo pedido a VENTAMATIC.
ATENCION: Las personas que se suscriban por 6 números de SPECTRUM TOP HITS antes del 30 de Septiembre de 1985, recibirán UN PROGRAMA-SORPRESA DE REGALO.
Fecha: ______
Nombre: ______
Apellidos: ______
Dirección: ______
Población: ______
Provincia: ______ D.P.: ______
Señalar con una cruz la forma de pago:
☐ Talón adjunto (sin gastos de envío).
☐ Giro Postal Nº ______ (sin gastos de envío).
☐ Contra-Reembolso (+ 200,– ptas. de gastos de envío).
☐ Tarjeta VISA / MASTERCARD / AMERICAN EXPRESS (+ 200,– ptas. de gastos envío),
Nº ______ Caduca: ______
Firma:
VENTAMATIC
¡Y ADEMAS! UN GRAN EXITO
CONTABILIDAD PERSONAL
20 Cuentas y 1.400 Asientos, 64 caracteres por línea, Compatible Microdrive/Cassette Todas las Impresoras e Interfaces
OFERTA ESPECIAL
Vale por 10% de DESCUENTO y un REGALO-SORPRESA en tu próxima compra de HARDWARE Y SOFTWARE directamente a VENTAMATIC este descuento no es aplicable a ejemplares de SPECTRUMANIA ni SPECTRUM TOP HITS.
reproducción sin valor

antes de guardar algo en ellos. Este proceso divide a la cinta en áreas definidas y ordenadas, para que la información guarde siempre un orden conocido. FORMAT*, con la información que aclara en cuál de las dos unidades localizar el *wafer* y el nombre que le daremos, hace el trabajo. El proceso es minucioso. Primero se mide la longitud total de la cinta en función del número de sectores capaces de guardar información. También dispone las cabeceras que delimitan a cada uno de esos sectores. Después se verifica el proceso y se crea un directorio, lugar en el que se guarda relación de los ficheros almacenados en el *wafer*.

Una serie de comandos facilita el acceso a las capacidades de esta unidad de almacenamiento masivo. CAT*"a:" visualiza en la pantalla el directorio del *wafer* insertado en la unidad A. Posteriormente podríamos pedirle que cargue un programa en la memoria del ordenador. LOAD*, complementado con a: (desde esa unidad) y el nombre del fichero lo hacen.

SAVE* realiza el proceso inverso, como hace suponer este familiar comando.

Para deshacernos del **Wafadrive** y su WOS (Wafadrive Operating System) tecleamos NEW, pero también perdemos el programa que hubiera en la memoria.

Pudiera ocurrir que algunos programas, particularmente los que incorporan código máquina, no funcionen cuando el **Wafadrive** se encuentre activo. La razón es que el mapa de memoria se reconfigura, reservándose espacio para almacenar los directorios de los *wafers* situados en las unidades A y B, los *buffers* para los canales de comunicación, las variables del BASIC Extendido, *buffers* para las operaciones de lectura/escritura, etc.

Resulta conveniente pedir el directorio nada más insertar un nuevo *wafer*. De esta manera queda almacenado permanentemente en la RAM y las tareas de carga y almacenamiento se ejecutan más rápidamente.



En la parte posterior están los conectores correspondientes al slot *de ampliación, al* interface ***Centronics*** *y al* ***RS232,*** *respectivamente.*

COMANDOS DEL WAFADRIVE

NEW
Pone a cero el sistema operativo WOS, deshaciendose de cualquier programa en BASIC que hubiera en la memoria.

NEW *
Inicializa el WOS, reservando unos 2 Kbytes de RAM para su funcionamiento.

NEW #
Limpia el area reservada a los programas en BASIC, sin poner a cero el WOS.

FORMAT *"d:nombre del wafer"
Formatea los wafers para ser utilizados. Cualquier cosa que hubiera almacenada en el resulta borrada.

FORMAT*"R";velocidad de transferencia"
Ajusta la velocidad de transmision del port RS232. Normalmente esta a 1.200 baudios.

CAT *"d:"
Visualiza el directorio del wafer insertado en la unidad especificada.

CAT #"d:"
Carga el directorio desde el wafer sin visualizarlo en pantalla.

OPEN # flujo,"canal"
Abre un flujo de datos determinado y lo envia a un canal especifico.

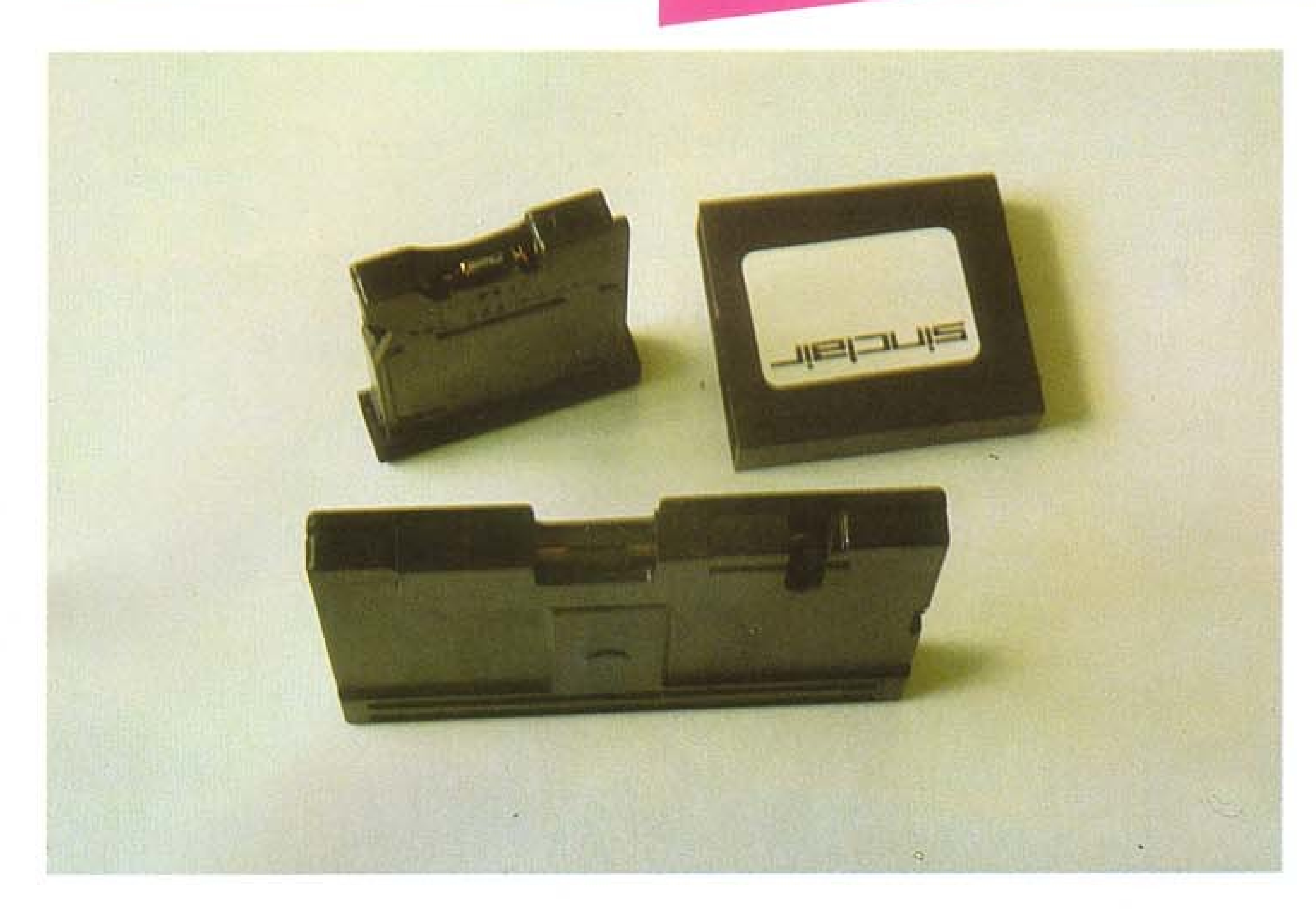
OPEN #* flujo,"port"
Similar al anterior, pero la asignacion tiene que ver con un port.

OPEN #* flujo,"d:nombre de fichero"
En este caso, si el fichero es nuevo, el canal es abierto como de salida, listo para la escritura. Si el fichero ya existiera, el canal se abre como de entrada, listo para leer.

CLOSE # flujo
Cierra un flujo previamente abierto con OPEN #. Es preferible utilizar CLOSE #* cuando el Wafadrive esta en operacion.

CLOSE #* flujo
Cierra un flujo abierto previamente por OPEN # u OPEN #*. Cerrar los flujos asignados a los canales de salida de ficheros hace que el fichero sea almacenado en el wafer, actualizandose el directorio.

SAVE *"d:nombre de fichero"LINE run
Almacena un programa en BASIC situado en la memoria. LINE es opcional y permite la autoejecucion del programa cuando se le vuelve a cargar en la memoria del ordenador.



El wafer *dispone de una tapa autodeslizante que protege a la cinta cuando está fuera de la unidad.*

```
SAVE *"d:nombre de fichero",comienzo,longitud, run
Almacena codigo maquina existente en la memoria. Igualmente existe la opcion
de que se autoejecute al cargarlo de nuevo.

SAVE #
Igual que SAVE *, pero no produce mensaje de error si existe un fichero del
mismo nombre, simplemente escribe encima de el.

VERIFY *"d:nombre de fichero"
Verifica un programa en BASIC o codigo maquina, contrastando lo grabado en el
wafer con el contenido de la memoria.

LOAD *"d:nombre de fichero"
Carga el programa pedido, independientemente de si esta escrito en BASIC o
codigo maquina. LOAD *, simplemente, carga el primer programa existente en el
directorio del wafer seleccionado por defecto (de mas indicaciones).

ERASE *"d:nombre de fichero"
Se deshace del fichero especificado, liberando los sectores asociados para un
uso posterior.

MERGE *"d:nombre de fichero"
Mezcla el programa en BASIC especificado con el previamente existente en la
memoria del ordenador.

MOVE *"d:nombre 1" TO "nombre 2"
Hace una copia del fichero que lleva el nombre 1, dandole el nombre 2.

PRINT #,flujo;"cadena",datos,variables
Envia datos numericos o cadenas a un flujo de salida abierto previamente.

INPUT #flujo,linea
Lee datos a partir de un canal de entrada abierto previamente, por medio del
flujo asignado.

INKEY$ # flujo;variables
Lee el proximo caracter por separado desde un canal de entrada abierto
previamente.

CLEAR *
Cierra todos los flujos que hubiera abiertos, al igual que los canales.

CLS *
Limpia la pantalla y ajusta los atributos de pantalla a los valores iniciales
BREAK
Es un comando directo que sirve para detener la ejecucion de un programa en
BASIC en cualquier momento.
```

De todos modos, existen muchas interesantes posibilidades de manejo del **Wafadrive** que salen fuera del objetivo de un artículo descriptivo como éste. El manual las detalla clara y sucintamente, pero con los detalles necesarios para sacar un buen partido. El distribuidor en nuestro país dispone de una traducción al castellano, que acompaña a la unidad.

Los *wafers* tienen un grosor ligeramente menor que los microcartuchos, sin embargo el tamaño es prácticamente el doble.

El mecanismo de protección de la cinta es más práctico e ingenioso que las modestas cajitas de **Sinclair.** Una tapa cubre la superficie expuesta cuando el *wafer* está fuera de la unidad y se desliza automáticamente cuando es insertado.

El **Wafadrive** es suministrado con un tratamiento de textos en *wafer,* que recuerda al popular **Tasword. Spectral Writer** es el nombre con el que ha sido bautizado este procesador de textos. Funciona con el **Spectrum** de 48 K y uno de sus principales méritos consiste en que sea posible visualizar hasta 64 caracteres por línea de texto escrito en la pantalla. La existencia de los *ports* **Centronics** y RS232 en el mismo **Wafadrive** permite que se puedan emplear impresoras de buena calidad para las copias en papel. El editor del texto en pantalla es prácticamente igual que el habitualmente utilizado para editar líneas de programa en BASIC.

El fabricante asegura que cada *wafer* tiene una vida media de 5.000 pasos por delante de la cabeza, aunque siendo cuidadosos es posible prolongarla. Esta cifra contrasta con otra sensiblemente menor adjudicada últimamente por los usuarios a los microcartuchos.

Para concluir diremos que el reto está en la calle. El usuario y los fabricantes de *software* tienen la palabra. El camino que abre el **Wafadrive** es de unas posibilidades incuestionables. Tal vez su mejor definición sería considerarlo en el punto intermedio entre el **Microdrive** y la unidad de *diskettes.*

ASI LO OYE TU ORDENADOR

- MUNDOS ANALOGICO Y DIGITAL
- LECTURA DE UNA SEÑAL
- LA TRAZA SONORA
- ALMACENAR Y REPRODUCIR



Ya puedes utilizar tu ordenador doméstico para introducirte en la tecnología de la grabación digital del sonido. Un sencillo programa permitirá al microordenador analizar un sonido, o incluso grabar un breve fragmento.

Todo sonido tiene dos componentes —frecuencia y volumen—, siendo capaz el oído humano de analizar e interpretar estas señales complejas que son, literalmente, vibraciones del aire que le rodea, admitiéndolas como sonidos reconocibles.

Visto de tal forma, el sonido es una señal analógica, es decir, que varía de una forma continua dentro de un amplio margen, siendo cada variación significativa. A diferencia de los sentidos humanos, los ordenadores no son capaces de reconocer esta clase de cambios, necesitando por el contrario una señal digital. En esta señal cada variación viene representada por un valor claramente distinto, un 0 o un 1, una presencia o una ausencia de señal.

Sin embargo, aunque el ordenador no puede interpretar directamente una señal sonora, es muy sencillo convertir la forma de onda analógica, por ejemplo de un fragmento musical, en información digital lista para ser utilizada.

En las aplicaciones prácticas esta técnica se sitúa al frente de la tecnología de grabación del sonido, que está evolucionando desde el tradicional almacenamiento en cinta (de la señal analógica) a sistemas computerizados con almacenamiento en disco magnético. Las ventajas son que una vez almacenado el sonido en forma digital es mucho más fácil modificarlo o combinarlo con otros sonidos, y una vez grabado es mucho menor el riesgo de «corrupción» debido a las limitaciones impuestas por el sistema de grabación.

MUSICA EN TU MICRO

Aunque actualmente esta tecnología solamente existe en unos cuantos sistemas caros y altamente sofisticados, un ordenador doméstico es capaz de ofrecer una aproximación a estas posibilidades.

Cada vez que cargas un programa (con LOAD) almacenado en una cinta de *cassette,* se está reproduciendo una señal que había sido almacenada digitalmente. Y como sabrás, si has escuchado alguna vez por el altavoz una cinta de ordenador, la señal produce además un tipo de sonidos que, aunque digitales, no admiten una interpretación con significado coherente al oído.

Pero con una adecuada programación (que comporta la utilización del código máquina), puedes utilizar tu sistema para poner música en el micro. Lo único que debes hacer es introducir en la máquina una señal analógica por el conector del *cassette,* y decir al ordenador cómo interpretarla, convirtiéndola en una señal digital.

Una vez que el ordenador ha hecho esto, puedes almacenar los números obtenidos en su memoria, e incluso visualizarlos en pantalla, como muestra el siguiente programa analizador de sonidos. Esto permite que un ordenador doméstico convierta la señal analógica de cualquier sonido grabado en una serie de números, que pueden utilizarse posteriormente para producir una traza gráfica en la pantalla, o ser almacenados en memoria, para la reproducción en otro momento, e incluso dentro de otro programa de ese sonido previamente digitalizado.

TRAZA SONORA

La primera opción que ofrece el programa es producir una representación gráfica, continuamente cambiante, de la música o sonido reproducidos en el magnetófono conectado al micro. Cuando se pone en marcha la cinta aparece en la pantalla una serie de líneas muy próximas cuya longitud depende directamente del sonido que se esté analizando: cuanto más alta es la frecuencia, más larga es la línea. Cuando la pantalla se llene, la traza desaparecerá y volverá a empezar a partir del lado izquierdo.

> **MICRONOTICIAS**
>
> **AJUSTE DEL VOLUMEN DE SONIDO**
>
> La posición normal del mando de volumen en el cassette puede no ser la ideal para una buena traza sonora. Si el magnetófono tiene una tecla de PAUSE, puedes servirte de ella para encontrar el nivel más adecuado. Haz correr la cinta y aprieta PAUSE mientras el ordenador está visualizando la traza. Obtendrás una traza plana, que muestra una señal que no cambia. A continuación sitúa el control de volumen justo en la posición en que el ruido de fondo empieza a romper la traza plana.

GRABACION DEL SONIDO

La segunda opción permite grabar el sonido introducido a través del *port* del *cassette,* aunque la cantidad que se puede grabar está muy limitada por las razones explicadas más abajo. La tercera opción permite reproducir el sonido grabado.

COMO FUNCIONA

Como el ordenador no puede interpretar directamente las variaciones de una señal analógica de sonido se le programa para ir asignando a la misma valores digitales. Lo que realmente hace es tomar muestras de la señal en el *port* del *cassette* a intervalos muy rápidos, miles de veces por segundo. La señal en el *port* del *cassette* sólo puede ser alta o baja, representada por un 1 o un 0; no hay valores intermedios como en una señal analógica. Pero como la velocidad de muestreo es tan rápida, la velocidad de cambio de la señal digi-

tal imita la forma de onda de la señal analógica.

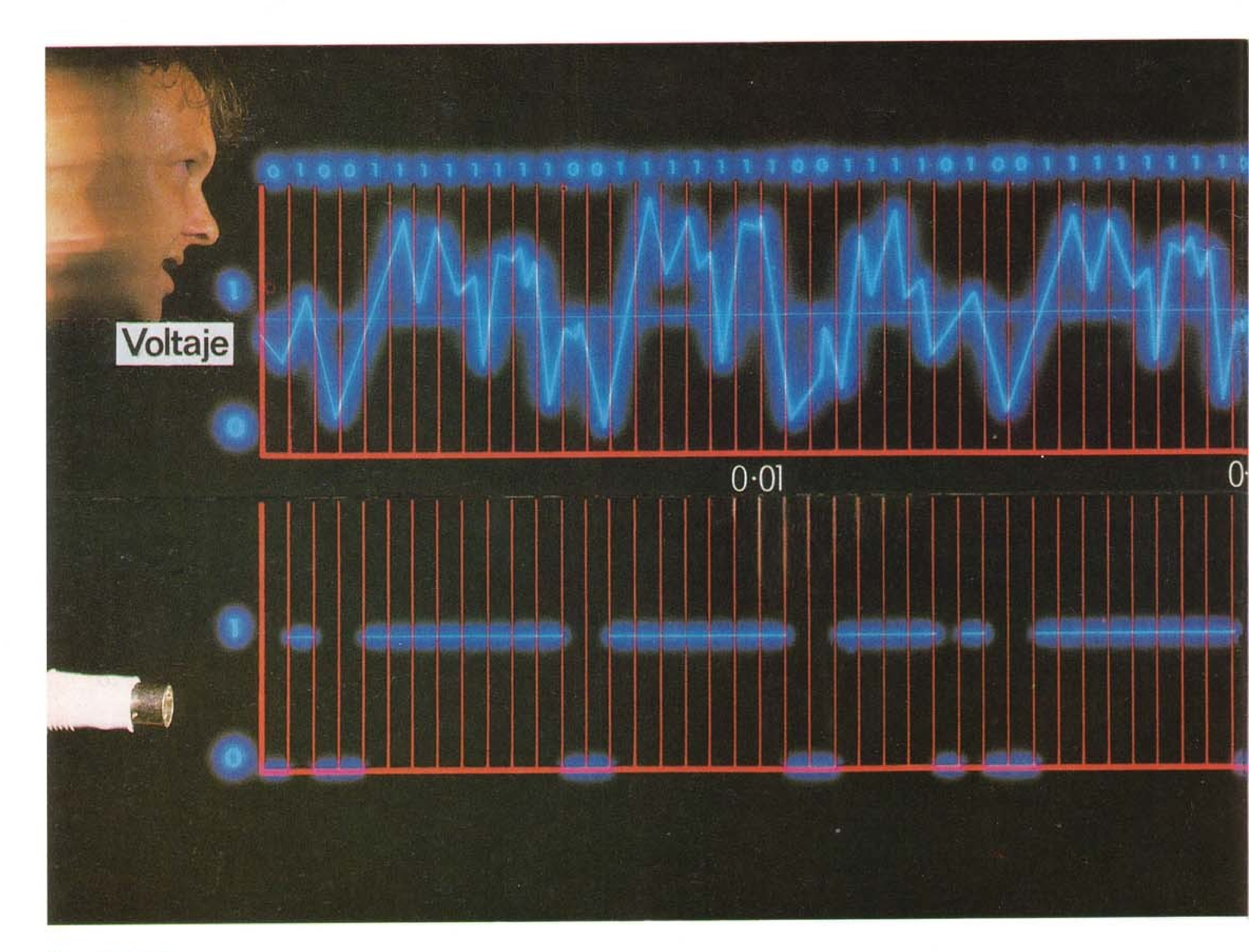
Imagina, por ejemplo, que se introduce una señal con una frecuencia de 256 Hz (es la nota Do central), en cuyo caso la señal alcanzará 256 máximos en un segundo y cada máximo permanecerá durante 1/512 de segundo. Si se realizan lecturas en el *port* a razón de 2000 lecturas por segundo, entonces cada máximo resultará leído durante aproximadamente cuatro tomas de muestras. A continuación la señal leerá un mínimo en otras cuatro muestras y así sucesivamente. De esta forma las variaciones de los valores digitales siguen la forma de onda de una manera más o menos precisa. Cuanto más rápido sea el muestreo, mejor captará el ordenador los cambios sutiles.

El programa utiliza los valores digitales que obtiene de las dos maneras descritas anteriormente. Para representar gráficamente la forma de onda, dibuja una figura mostrando cuántos bits altos encuentra por unidad de tiempo, es decir, la frecuencia global del fragmento de música de que se trate. Aquí también, cuanto más frecuente sea el muestreo y más frecuente el dibujo, más precisa será la representación gráfica.

La grabación se realiza de una manera parecida. El ordenador cuenta ocho muestras y las almacena como un byte de memoria. A continuación se repite el proceso para las siguientes ocho muestras, que se colocan en la siguiente posición de memoria. Esto sucede tan rápidamente que se requiere una gran cantidad de memoria: por ejemplo en el **Spectrum,** se requiere toda la memoria disponible para almacenar unos ocho segundos de sonido.

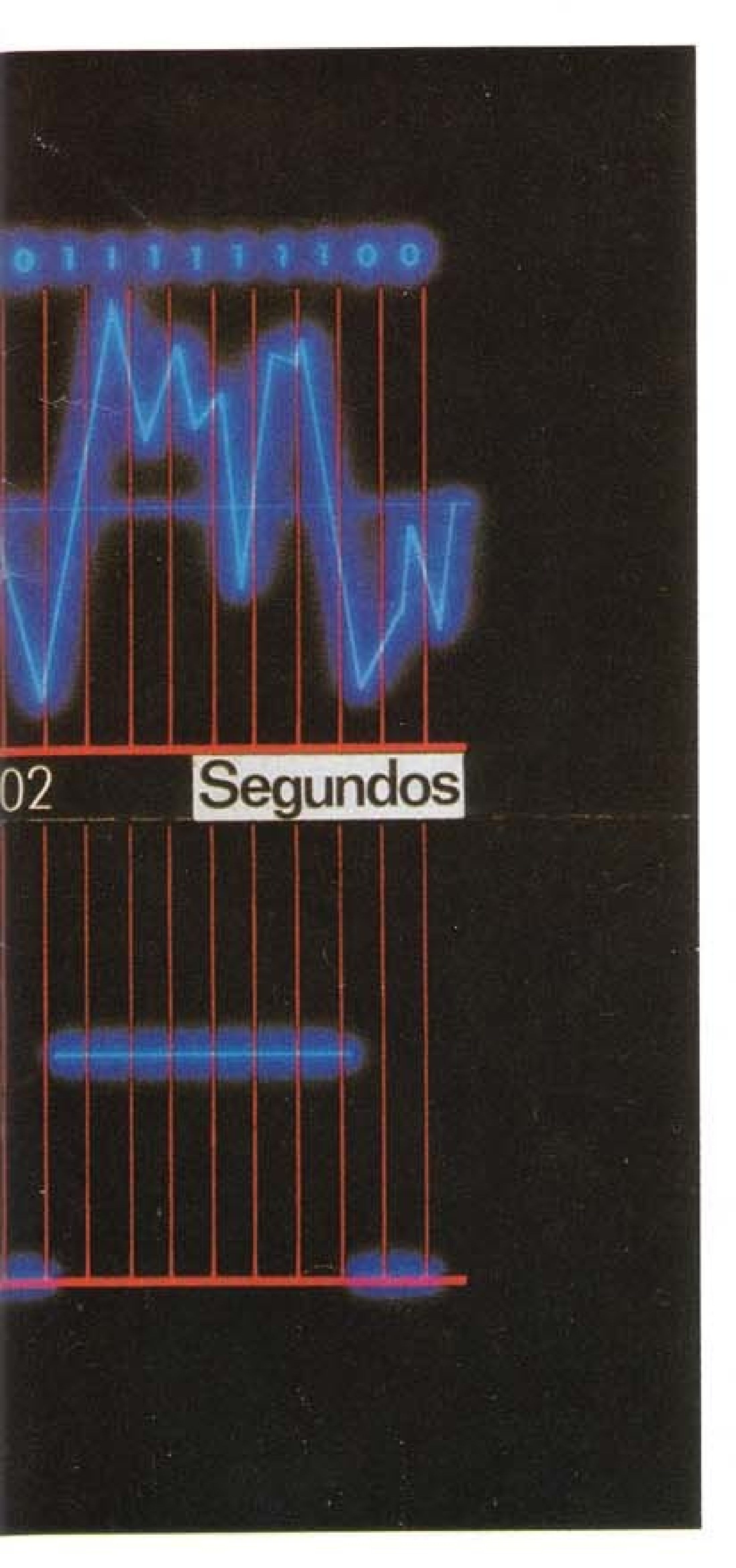
El proceso de reproducción es exactamente el inverso de éste, haciendo que la información almacenada salga por la toma de sonido para

Una señal analógica que entra al port *del* cassette *se muestrea 2000 veces por segundo. Cuando su valor está por encima de la tensión umbral se detecta y se graba un 1. Las señales por debajo de este nivel dan un 0. La traza producida digitalmente de esta manera imita, en efecto, la forma de onda analógica del sonido.*



representar las vibraciones que estaban presentes en el *port* del *cassette* durante la grabación. ¡Pero debido a las limitaciones del equipo el resultado final no es precisamente alta fidelidad!

Este programa se compone de dos partes: un listado en BASIC para establecer las rutinas y la representación en pantalla, y una rutina en código máquina para leer el *port* del *cassette*. Teclea la primera parte y almacénala (utilizando SAVE "ANALIZADOR" LINE 5) y a continuación teclea la segunda parte, que contiene el código máquina en forma de DATA que es POKEado en la memoria por medio de la línea 30. La línea 80 contiene una comprobación para cerciorarse de que se han introducido los números correctamente. Ejecuta el programa (RUN) para introducir el código máquina, y a continuación escribe SAVE "ANALIZADOR" CODE 65368, 109, inmediatamente después del programa en BASIC.



Carga el programa en BASIC (con LOAD), que autoarrancará a partir de la línea 5 y cargará los datos del código máquina en la parte alta de la memoria después de establecer un nuevo RAMTOP. Cuando se completa la carga aparece un menú con tres alternativas.

La primera opción pedirá que conectes la salida de auricular del magnetófono a la correspondiente conexión para auricular del **Spectrum** y a continuación reproduce algún sonido grabado en la cinta. Cuando posteriormente pulsas cualquier tecla se genera continuamente un gráfico de frecuencia en función del tiempo. Para volver al menú principal pulsa la M, o la F si quieres congelar la imagen. Si se pulsó F, volverá a generarse una nueva imagen con la pulsación de cualquier tecla.

Si se elige la opción dos, nos invitará a pulsar cualquier tecla para empezar a grabar, y nos informará de cuándo termina la grabación, antes de producirse un retorno automático al menú principal. Se puede grabar cualquier sonido, pero los sonidos cortos y agudos resultan mejor ya que, debido a las limitaciones del **Spectrum,** el sonido que se escucha en reproducción cuando se elige la opción tres va más lento que su velocidad natural, resultando filtradas la mayor parte de sus componentes de baja frecuencia. El *cassette* no es capaz de distinguir las frecuencias por debajo de un cierto valor umbral relativamente alto. Como resultado, la grabación resulta pastosa y con un tono más profundo.

Debido a la enorme cantidad de memoria necesaria para almacenar lo que al fin y al cabo es una pequeña cantidad de sonido de mala calidad, no tiene mucho sentido intentar incorporar el sonido grabado en otros programas. Sin embargo, si deseas experimentar, es posible hacerlo introduciendo SAVE "sonido" CODE 26000, 39360. A continuación almacena el código máquina para ejecutar el programa, introduciendo SAVE "ejecutor" CODE 65440, 40.

Cuando quieras utilizarlo con un programa propio, haz una inicialización del RAMTOP introduciendo CLEAR 25999. Seguidamente carga los datos del sonido, seguidos del ejecutor, con LOAD ""CODE : LOAD"" CODE.

Para escuchar el sonido introduce RANDOMIZE USR 65440. Recuerda que sólo dispones de alrededor de unos 3K de memoria para escribir el resto de tu programa, lo cual no es mucho, a menos que seas un programador muy eficiente en lenguaje máquina.

Teclea

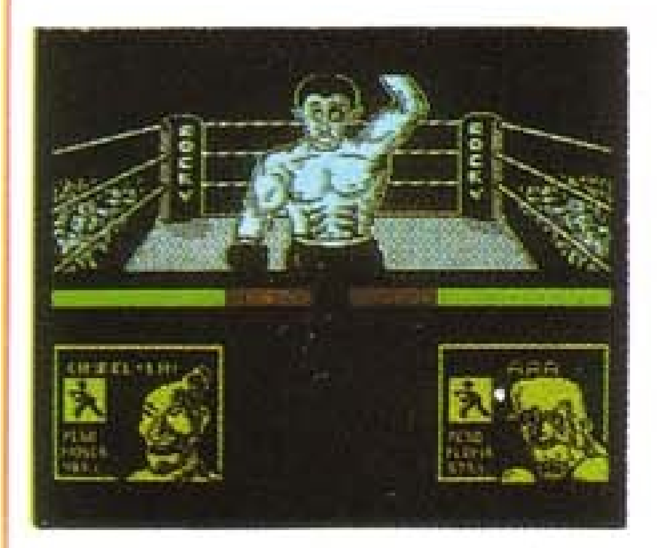
```
5 CLEAR 25999: LOAD ""CODE
10 GO SUB 200: GO SUB 500
20 IF INKEY$="" THEN GO TO
   20
21 LET C=CODE INKEY$: IF
   C<49 OR C>51 THEN GO TO
   20
22 GO SUB 30: GO SUB 200
24 IF C=49 THEN GO TO 100
25 IF C=50 THEN GO TO 600
26 IF C=51 THEN GO TO 700
30 FOR N=30 TO 50 STEP 3:
   BEEP .01,N: NEXT N:
   RETURN
100 CLS : GO SUB 1000: PRINT
   AT 12,8; BRIGHT 1;"
   PULSA UNA TECLA "
101 IF INKEY$="" THEN GO TO
   101
102 BEEP .1,10
104 CLS : GOSUB 150: GOSUB
   800
105 FOR X=0 TO 255: PLOT
   X,0: DRAW 0,USR 65368
110 IF INKEY$="m" THEN GO
   SUB 30: GO TO 10
111 IF INKEY$="f" THEN GO
   SUB 801: GO TO 140
130 NEXT X: CLS : GO SUB 150:
   GO SUB 800: GO TO 105

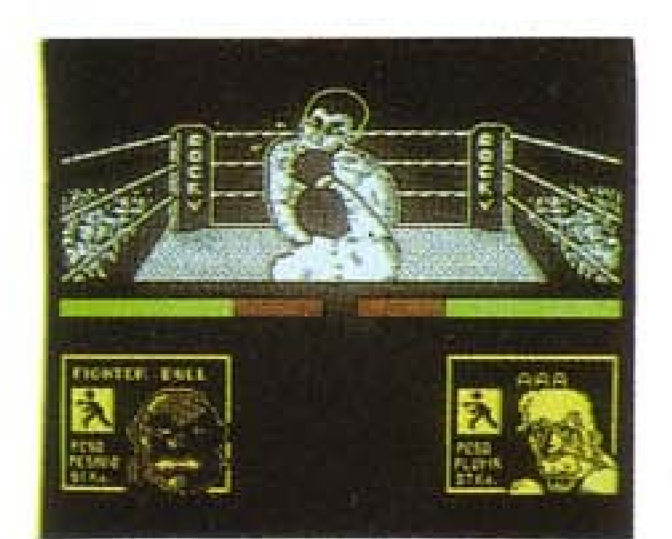
140 PRINT AT 0,0;"□□□□
   □□□□□□□□□□
   □□□□□□□□□□
   □□□□□□□□□":
   BEEP .1,40: PAUSE 50:
```

```
    IF INKEY$="" THEN GOTO
    140
141 BEEP .1,10: CLS : GO TO
    104
150 PRINT AT 0,0; BRIGHT
    1;"□□PULSA(M) PARA
    VOLVER AL MENU□□":
    RETURN
200 BORDER 5: PAPER 5: INK
    0: CLS : RETURN
500 PRINT AT 0,5; PAPER 2;
    INK 7;"□□MENU DEL
    ANALIZADOR□□"
510 PRINT AT 5,7;"1.BARRAS
    FREC.";AT 7,7;
    "2.REGISTRA SONIDO";
515 PRINT AT 9,7;
    "3.REPRODUCE SONIDO"
520 PRINT AT 15,5; PAPER 4;
    "PRESIONA (1) (2) O
    (3)": RETURN
600 PAUSE 20: GO SUB 1000:
    PRINT AT 13,0;"Pulsa
    una tecla para GRABAR"
605 IF INKEY$="" THEN GO TO
    605
606 BEEP .05,20: CLS :
    PRINT AT 10,10;"UN
    MOMENTO": RANDOMIZE USR
    65408: BEEP .1,30: CLS
    : PRINT AT 10,6; BRIGHT
    1;"□GRABANDO□": PAUSE
    300: GO TO 10
700 RANDOMIZE USR 65440: GO
    TO 10
800 PRINT AT 1,0; PAPER 4;"
    □PULSA(F) PARA CONGELAR
    IMAGEN": RETURN
801 PRINT AT 1,0; PAPER 4;"
    PULSA UNA TECLA PARA
    CONTINUAR□": RETURN
1000 PRINT AT 4,2;"Conecta
    la toma de auricular";
    AT 6,0;"del cassette a
    tu SPECTRUM";AT 8,0;"y
    reproduce el sonido":
    RETURN
```

```
10 CLEAR 26000: RESTORE :
   LET t=0: LET x=65368
20 FOR n=1 TO 108
30 READ a: POKE x,a
40 LET t=t+a
50 LET x=x+1
60 NEXT n
70 IF t=12721 THEN PRINT
   "BIEN": STOP
80 PRINT "ERROR EN LAS
   DATA": STOP
90 DATA 14,64,175,8,17,
   208,7,219,254,230,64,
   185,40,7,62,64,169,79,
   8,60,8,29,32,239,175,
   178,40,5,30,255,21,24,
   230,8,203,63,6,0,79,201
100 DATA 243,33,144,101,17,
   80,255,6,7,219,254,203,
   119,32,2,203,254,203,
   62,16,244,35,125,187,
   32,237,124,186,32,233,
   251,201,243,33,144,101,
   17,80,255,6,8,203,70,
   40,4
110 DATA 62,0,211,254,62,
   255,211,254,203,14,16,
   240,35,125,187,32,233,
   124,186,32,229,251,201
```

NO TE RESISTAS A TENER UN ORIGINAL DINAMIC





SPECTRUM PLUS Y 48 K
1.800 ptas.



Simulación tridimensional de un combate de boxeo. Lucha contra: CIMBEL-LIN; TED MATARE, JANSEN SINO y FIGHTER para conquistar el título del mundo. Visualización en pantalla de los niveles de fuerza y fichas técnicas de los combatientes. Programa por ARMID SOFT. Gráficos realizados por DINAMIC.

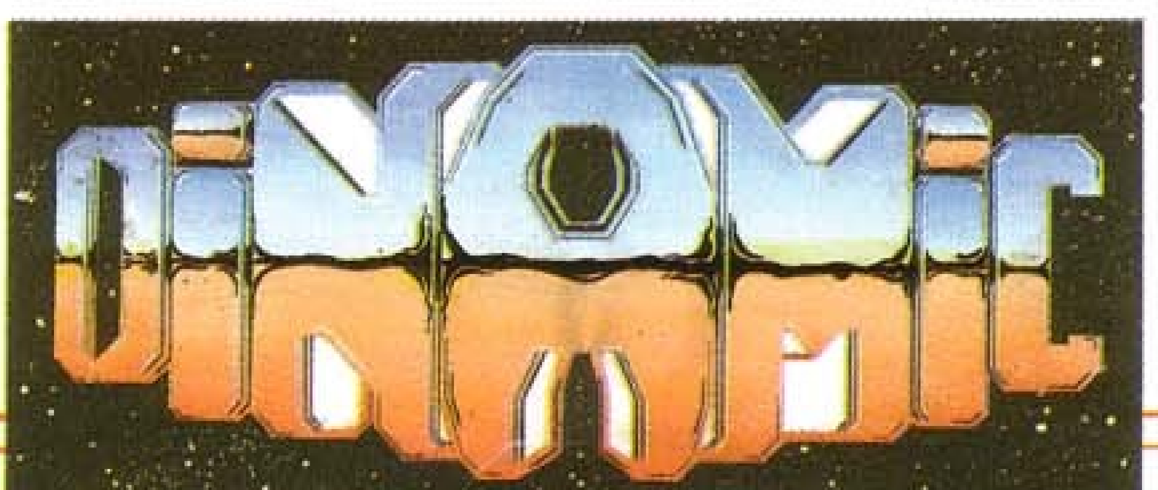






SPECTRUM PLUS Y 48 K
2.100 ptas.

Durante más de 3.000 años, los más famosos exploradores han intentado profanar el templo de ABU SIMBEL. Momias, arañas, flechas envenenadas, etc., se encargan de su custodia. Llegar hasta la cámara mortuoria es inaccesible; salir con vida, imposible; todos los que intentaron llegar, jamás regresaron. Ahora, Johny Jones, nuestro héroe, lo va a intentar.



SOFTWARE ESPAÑOL

Tiendas y Distribuidores: 447 34 10
Pedidos contrarreembolso a:
«Mansion Dinamic»
Tel.: (91) 715 00 67
Tilos 2, 21. Monteprincipe
Boadilla del Monte
(MADRID)

PROGRAMADORES
¿Quieres ver tu programa lanzado al mercado como ROCKY y PROFANATION? ¿Quieres entrar en el «TOP 10» español e inglés? No lo dudes, ponte en contacto con nosotros.

CUIDADO DE CINTAS Y DISKETTES

- PROTEGER LA INFORMACION
- HACER COPIAS DE SEGURIDAD
- INDEXAR
- ENVIO POR CORREO CINTAS Y DISCOS

Incluso si tú eres una de esas personas que guardan su información importante en la parte posterior de un sobre ya utilizado, cuando guardes ésta en cinta o *diskette,* desde luego, no te merecerá la pena ser desorganizado.



Los sistemas de almacenamiento basados en cintas o discos magnéticos hacen posible que el usuario de ordenadores personales pueda guardar grandes cantidades de información, o miles de programas, de una manera extremadamente compacta.

Pero la eficiencia que proporcionan los sistemas de almacenamiento magnético es también un potencial punto débil. Debido precisamente a que una diminuta cinta o *diskette* pueden contener tanta información, cualquier daño que se les produzca se convierte en un resultado desastroso. Aquí, el soporte magnético es un material particularmente vulnerable.

SEGURIDAD

Obviamente, lo más importante es asegurarse de que una vez depositada la información en la cinta o *diskette* ésta permanecerá allí, pudiendo ser recuperada siempre que lo deseemos. Pero existen dos clases de daños que pueden afectar a este tipo de soporte: el físico y el magnético.

El daño físico puede ser cualquier cosa, desde doblar un *diskette* o triturar una cinta, hasta acumular suciedad en su superficie. Esto se previene fácilmente, protegiéndolos siempre que no se estén utilizando. Se deben guardar en su sobre e introducirlos en una caja adecuada para *diskettes.* Las cintas también deberán meterse dentro de sus cajas y, preferiblemente, emplear uno de esos módulos de plástico que hay para apilar un buen número de cajas. Finalmente, cabe recomendar su almacenamiento lejos del calor, la humedad y el polvo.

Tampoco se debe tocar la superficie magnética de un *diskette* o el trozo de cinta que asoma por un *cassette.* También es un buen hábito rebobinar las cintas hasta su comienzo, una vez que hemos acabado de utilizarlas; esto significa que dejamos al descubierto la parte menos vulnerable de la misma. Asimismo logramos que la cinta esté disponible para su inmediato uso en cualquier momento. No es conveniente dejar olvidados los *diskettes* y las cintas en sus correspondientes unidades durante largos períodos de tiempo después de utilizar el ordenador.

Existen determinados campos magnéticos que tienen fuerza suficiente para alterar el contenido de una cinta o *diskette.* Estos proceden de diversos electrodomésticos habituales en el hogar. Por ejemplo, los altavoces contienen potentes imanes, y algunos motores eléctricos también. Por lo general, cintas y *diskettes* se deben mantener bien alejados de cualquier equipo eléctrico, incluyéndose la televisión.

MAS SEGURIDAD

El accidente puede tener lugar, por lo que merece la pena hacer copias de seguridad o tomar algún tipo de precauciones extraordinarias, especialmente para las grabaciones más importantes.

Por lo general, es una buena idea grabar dos versiones de todo, incluso en la misma cinta o *diskette.* Existen menos probabilidades de que se deterioren dos que una sola. Pero si el fichero es realmente importante haz otra copia en distinto lugar. Se almacenará en otro sitio y preferiblemente no se utilizará. Otro consejo es guardar en cinta copias de los programas existentes en *diskette* —la cinta es menos delicada.

Proteje tus cintas contra la sobreescritura quitando las lengüetas

posteriores del *cassette*. Los *diskettes* pueden ser protegidos igualmente con sólo pegarles una etiqueta adhesiva en las muescas laterales de la funda.



SIGUE LA PISTA

Las cintas y *diskettes* pueden haberse utilizado hace mucho tiempo, por lo que resultará difícil recordar, con exactitud, lo que hemos depositado en ellos. Una regla general consiste en etiquetar las cosas tanto como podamos. Si tu economía te lo permite, no está de más almacenar cada fichero por separado, en cintas cortas, o reservando *diskettes* para ficheros relacionados entre sí.

Proporciona a cada fichero un nombre claro y único (dentro de los límites que permita tu sistema). En particular, si tienes varios desarrollos de un mismo programa, dale a cada uno un nuevo nombre (o número). Escribe los nombres de los ficheros en la etiqueta del *cassette*, así como en la tarjeta que va en la caja, pues ambas deben ir asociadas en caso de separación. No escribas en las etiquetas de un *diskette* con un lápiz duro porque te puedes cargar el *diskette*. Es preferible emplear un rotulador de punta de fieltro. Un libro que actúe como índice de ficheros será de gran ayuda. En él puedes indicar todos los nombres, dónde está el fichero y cualquier otra anotación. Merece la pena incluir una sentencia tipo REM en el fichero, en la que se anota la fecha y descripción.

ENVIO DE FICHEROS

Las cintas y *diskettes* son una manera muy conveniente de enviar información o programas por correo. Las cintas son razonablemente fuertes y viajan bien. Sorprendentemente, lo delicado y más propenso a rotura es la cajita, por lo que si la evitas, además ahorrarás peso. Aunque difíciles de encontrar, existen carteritas para envío por correo, pero otro tipo de bolsas almohadilladas son perfectamente adecuadas.

El mayor riesgo que corren los *diskettes* es el de ser doblados. Por ello un bocadillo hecho con dos láminas fuertes evita este problema. Es muy importante disponer un mensaje en parte visible del paquete, en el que se lea algo así como: MATERIAL MAGNETICO, MANEJESE CON CUIDADO. Para mayor seguridad, el correo certificado suele ser una buena solución.

LOS MEJORES DE INPUT

*Hemos pensado que es interesante disponer de un **ranking** que ponga en claro, mes a mes, cuáles son los programas preferidos de nuestros lectores. Para ello, es obligado preguntaros directamente y tener así el mejor termómetro para conocer vuestras preferencias. Podríamos haber recurrido a otros métodos de muestreo, pero éste no deja lugar a dudas.*

*El resultado de las votaciones será publicado en cada número de **INPUT.***

Entre los votantes sortearemos 10 cintas de los títulos que pidáis en vuestros cupones.

***Nota:** No es preciso que cortéis la revista, una copia hecha a máquina o una simple fotocopia sirven.*

Enviad vuestros votos a:

LOS MEJORES DE INPUT
Alberto Alcocer, 46-4.° B
28016 Madrid

ELIGE TUS PROGRAMAS

Primer título elegido

....

Segundo título elegido

....

Tercer título elegido

....

Programa que te gustaría conseguir

....

¿Qué ordenador tienes? ....

Nombre ....

Apellidos ....

Dirección ....

....

Teléfono de contacto ....

INPUT SINCLAIR

COMO TRABAJA EL SISTEMA OPERATIVO

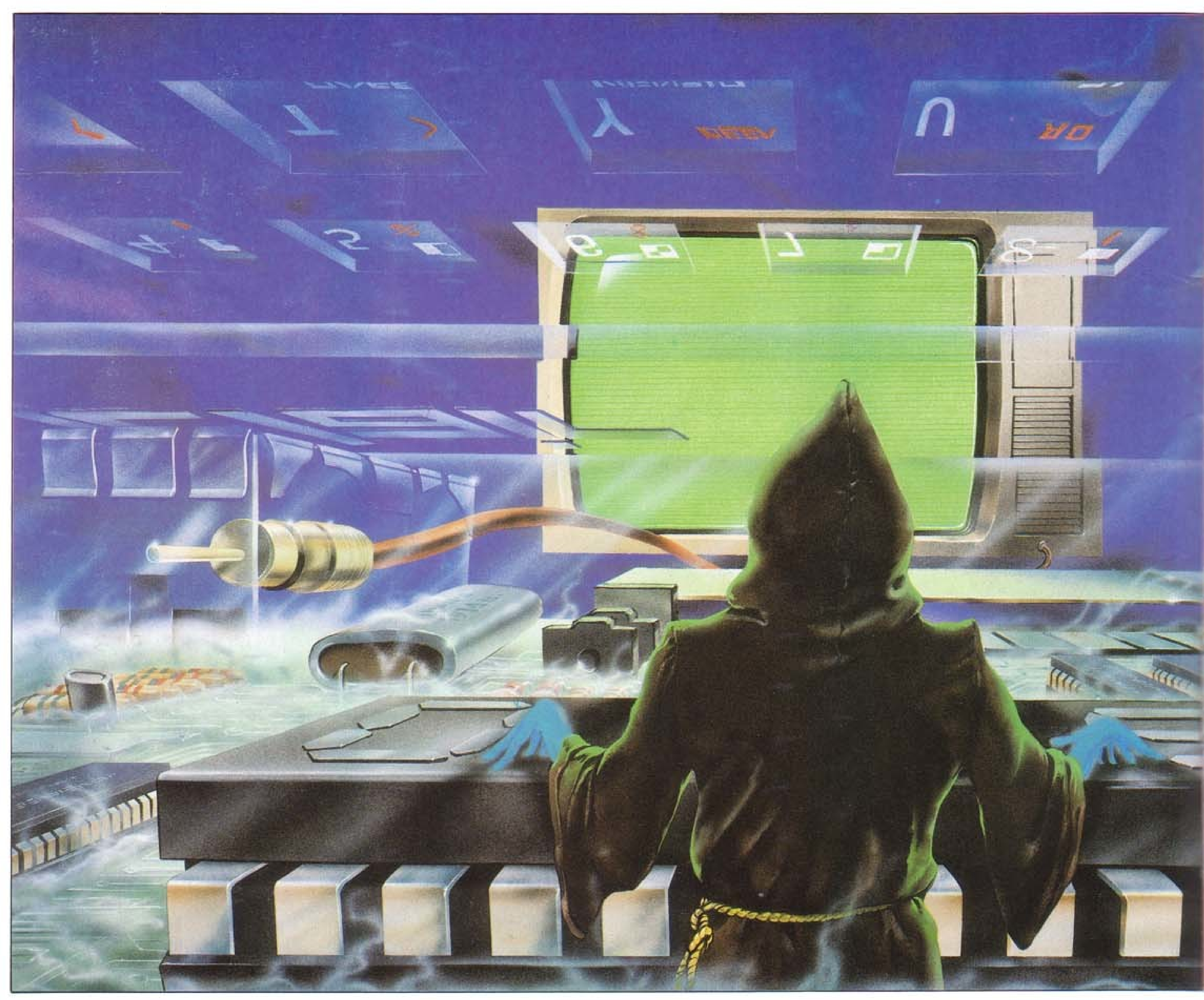
- ¿QUE ES EL SISTEMA OPERATIVO (S.O.)?
- TRABAJAR POR ENCARGO
- ENTENDER EL BASIC
- RUTINAS DE INTERES

Todos los ordenadores trabajan y entienden un solo lenguaje: el código máquina. Consiste éste en una interpretación instantánea de las variaciones de tensión eléctrica en base a las cuales opera el ordenador.

El usuario puede representarlo de una forma sencilla, con un simple conjunto de letras y números fáciles de recordar, o de una forma complicada, con una lista de ceros y unos casi incomprensible. Cualquiera que sea la forma en que esté representado, pocas personas pueden trabajar cómodamente con el código máquina, resultando necesario un medio de enlace o *interface* con el ordenador, que permita la comunicación a un nivel más sencillo. Este *interface* es el Sistema Operativo (S.O.), un paquete de *software* que gobierna las funciones del ordenador y hace posible que te comuniques con la máquina.

Independientemente del lenguaje que utilices, es seguro que éste se servirá de un sistema operativo para controlar al ordenador.

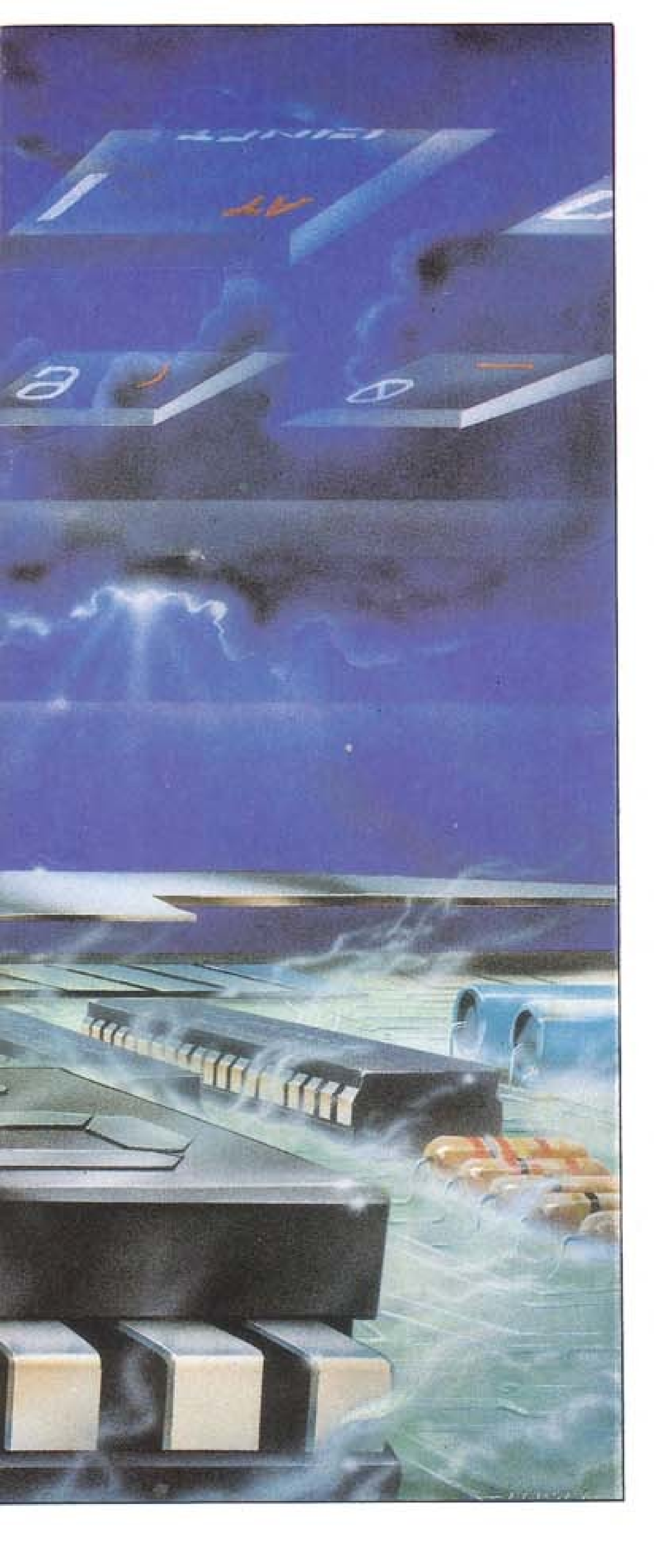
Aunque no es necesario entender en detalle cómo funciona el S.O., su



conocimiento te permitirá aumentar tu habilidad como programador y obtener mayor provecho del microordenador. Hasta dónde penetrar en el S.O. depende del diseño de tu ordenador. Dentro de estos límites, este artículo pretende familiarizarte con el S.O. de tu modelo y mostrar cómo puede utilizarse parte de la amplia colección de rutinas de actuación rápida que incorpora.

¿QUE ES EL S.O.?

El S.O. no es más que un sofisticado programa en código máquina que permite al procesador, incluido en un determinado ordenador, responder a tus instrucciones. Controla las interacciones de la máquina con el mundo exterior, la pantalla, el teclado, el generador de sonidos y otros *ports* de Entrada y Salida. El S.O. garantiza también que las posiciones de memoria están adecuadamente asignadas y que el procesador y el usuario utilizan la memoria de una manera eficiente. Cuando se conecta el ordenador, el S.O. inicializa ciertas rutinas de la ROM y carga los registros internos y los punteros con determinados valores iniciales, todo ello antes de enviar al usuario la señal de que está preparado para seguir adelante. En el caso de los microordenadores como del que tratamos aquí, el sistema operativo se ocupa normalmente de manejar los comandos del BASIC, pero también puede manejar otros lenguajes.



La mayor parte de la actividad de un S.O. está relacionada con el ajuste y comprobación de punteros. Por ejemplo, es importante que el procesador sepa dónde empiezan y terminan los programas en BASIC. Cuando se está editando el programa, va cambiando su tamaño y los punteros han de ser actualizados.

El mismo principio se aplica al ajuste y mantenimiento de espacio para almacenamiento de variables y, especialmente, de cadenas alfanuméricas. Cuando se DIMensiona una cadena hay que reservar una zona de memoria para ella y posteriormente ha de borrarse cuando la cadena ya no se necesite. Es esencial que el S.O. gestione la memoria de una forma eficiente, de lo contrario los programas o cadenas largos ocuparán toda la memoria. Esta gestión de memoria se llama mantener la casa en orden, y una buena gestión de memoria es una de las características que distinguen a un ordenador eficiente.

El S.O. incorpora una serie de rutinas de *software* que han sido convenientemente dispuestas para poder ser reclamdas por el programador o por el propio S.O. El programador de BASIC no necesita la mayoría de estas rutinas, ya que las funciones que realizan están contenidas en instrucciones normalmente introducidas cuando se utiliza el BASIC. Un buen ejemplo de esto es la palabra clave INPUT del BASIC. Esta palabra realmente utiliza varias subrutinas del S.O., incluyendo entrada, ajuste del canal, exploración del teclado y transferencia del *buffer,* pero el usuario del BASIC puede reclamarlas a todas utilizando una sola instrucción.

TRABAJANDO POR ENCARGO

Cada vez que se da una instrucción al ordenador, el S.O. activa a la sección adecuada del mismo. Por ejemplo, si se pulsa una A en el teclado, es el S.O. quien da instrucciones al microprocesador para que saque una A por la pantalla. Para hacer esto, detecta que se ha pulsado una tecla. En algunas máquinas, entre ellas el **Spectrum,** el S.O. explora el teclado regularmente para detectar las pulsaciones de las teclas. En otras máquinas, cada pulsación de tecla envía una señal de interrupción al S.O., que explora entonces el teclado para localizar la tecla que ha sido pulsada.

Para reaccionar ante la pulsación de una tecla, el S.O. va a la subrutina que se utiliza para sacar por pantalla un carácter. Esta es una de las muchas subrutinas en código máquina que se ejecutan muy rápidamente. Si se pulsa RETURN o ENTER, el S.O. se dirige a una subrutina de "nueva línea" y la ejecuta, disponiendo al cursor en el principio de una línea nueva.

ENTENDER EL BASIC

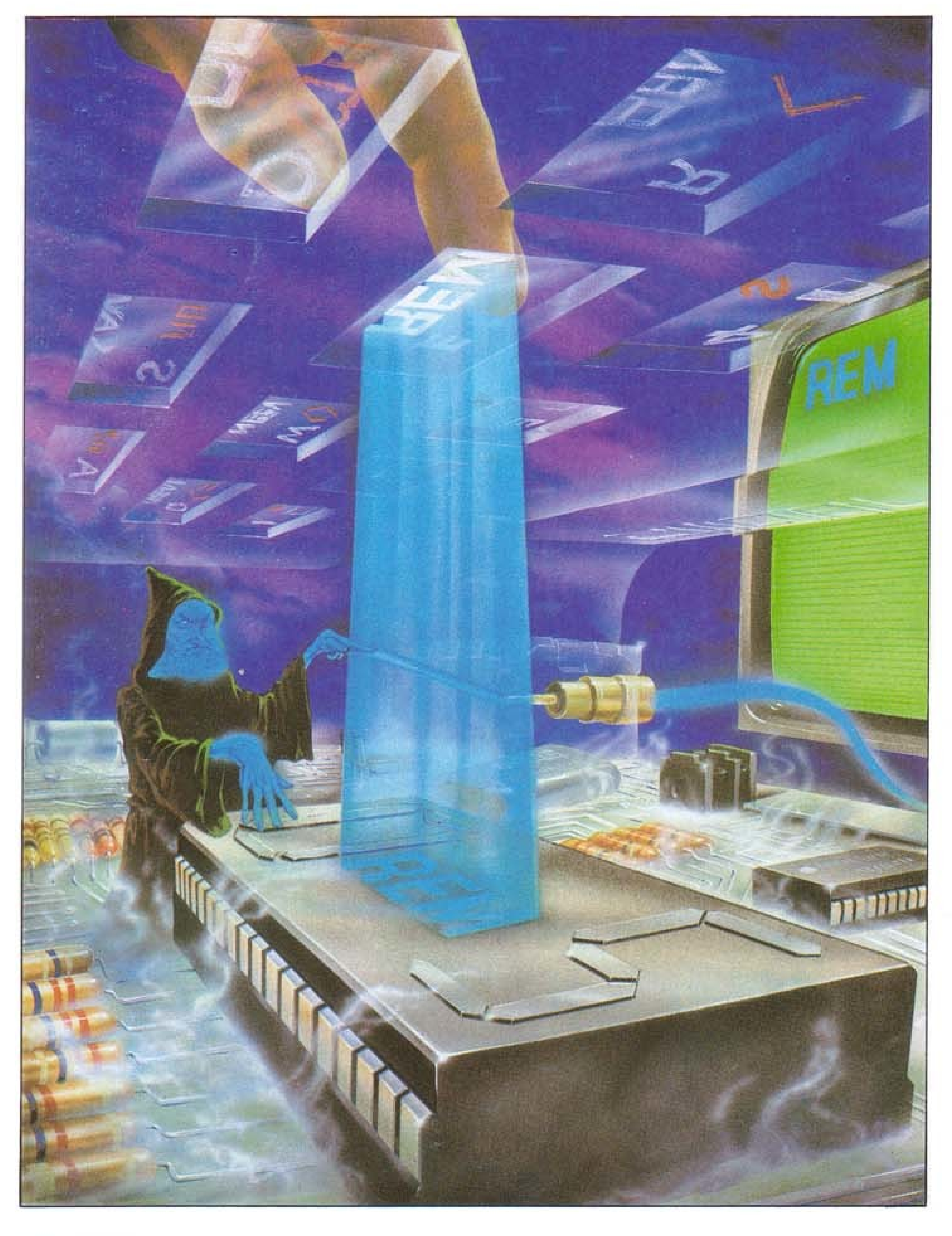
Cuando se ejecuta un programa (RUN) en un lenguaje, tal como el BASIC, las instrucciones almacenadas en la memoria son interpretadas, es decir, traducidas a código máquina, que es lo que entiende el ordenador. Al mismo tiempo, el código ajusta los registros para que se pueda acceder a las subrutinas del S.O. De esta forma las mismas subrutinas que permiten que el ordenador responda a las entradas introducidas

por el usuario a través del teclado, por ejemplo, son utilizadas por el S.O. para ejecutar los programas en BASIC.

¿Entonces por qué —podríamos preguntar— resulta el BASIC tan lento comparado con el código máquina, habida cuenta de que utiliza estas subrutinas rápidas? La respuesta es que el lenguaje ha de ser interpretado a partir de palabras clave, tales como PRINT y otros símbolos antes de que se pueda acceder a las subrutinas del S.O. El largo tiempo necesario para traducir el BASIC es la razón por la que muchos juegos de marcianitos y algunos programas que han de tener una respuesta rápida, se escriben en código máquina.

El tiempo que se tarda en escribir en lenguaje máquina se estima a menudo en unas diez veces el tiempo que se tardaría en escribir un programa en BASIC que hiciera lo mismo, pero el resultado puede ser cincuenta veces más rápido. Sin embargo, a menos que sea absolutamente necesario utilizar el código máquina, la mayor parte de la gente utiliza el BASIC por la facilidad para escribir programas.

Sin embargo sería de gran utilidad para los programadores en BASIC el poder acceder a las subrutinas del S.O. directamente, en vez de hacerlo a través del intérprete, y aprovecharse así de sus muchas subrutinas eficientes. Por desgracia, la mayoría de los micros domésticos no poseen un S.O. al que se pueda acceder directamente desde el BASIC.



El **Spectrum,** muy restrictivo en este aspecto, no permite cambiar el contenido de los registros, excepto en código máquina, cambio que es fundamental para dirigir al S.O. hacia las distintas subrutinas.

Pese a la diversidad de accesos que presentan las distintas máquinas, hay algunos hechos que los usuarios de los sistemas, incluso de los más restrictivos, pueden aprovechar para acortar la codificación o para resolver problemas o errores.

Se accede a las subrutinas del S.O. con la palabra clave USR, que se utiliza también para ejecutar una subrutina en código máquina colocada por el usuario en algún lugar de la memoria. Ensaya con el comando directo RANDOMIZE USR 0. Con esto se produce una inicialización completa del sistema. Para programas en BASIC, hay algunas variables útiles del sistema que pueden ser POKEadas.

POKE 23561, con un número entre 1 y 255, fija el tiempo de retardo antes de que las teclas se autorrepitan. La posición normal es POKE 23561,35. Análogamente, POKE 23562,5 es el intervalo normal de autorrepetición, pero se puede variar a voluntad entre 1 y 255. Estos cambios pueden ser útiles en programación de juegos o en cualquier programa que requiera que el usuario responda por medio del teclado.

Las direcciones 23606 y 23607 contienen la dirección del principio de la tabla de los diagramas de puntos para el conjunto de caracteres. Si se escribe POKE 23606,8 (el byte bajo), el puntero se desplaza un carácter en la tabla, de forma que para cualquier carácter que se pulse, aparecerá el que va a continuación. Introduce este POKE y luego pulsa 1,2,3,4. Observa que lo que aparece es 2,3,4,5, lo que proporciona un método sencillo para codificar programas

en BASIC, y disuadir a cualquiera de que los lea. Un programa codificado de esta manera se ejecutará normalmente, pero la salida por pantalla será muy confusa. Si ahora se pulsa POKE 23607,0 (el byte más alto), el puntero apunta ahora al principio de la ROM, de forma que los caracteres aparecen como un galimatías sin sentido.

La posición de memoria 23658 proporciona el único medio para forzar CAPS LOCK cuando se está ejecutando un programa en BASIC. POKE 23658,0 cambia el modo mayúsculas a minúsculas y POKE 23658,8 permite sólo mayúsculas.

PLOT especifica en el **Spectrum** una posición absoluta a partir del origen de la pantalla, mientras que DRAW está referido al último punto especificado. A veces se requiere un DRAW absoluto, lo cual se consigue con las posiciones de memoria 23677 y 23678. Por ejemplo, para situar un punto en el centro de la pantalla, introducimos PLOT 128,85. Ahora supón que deseas trazar una línea desde este punto hasta la parte superior derecha de la pantalla de coordenadas absolutas (255,175). Efectivamente, DRAW 255,175 especifica un punto fuera de la pantalla, pero DRAW 255-PEEK 23677,175-PEEK 23678 nos dará la línea deseada. Restando PEEK 23677 de la coordenada X y PEEK 23678 de la coordenada Y, se vuelve a coordenadas DRAW relativas.

La forma más inmediata de utilizar el **Spectrum** como reloj es el contador de cuadros de la pantalla del televisor. Tres posiciones de memoria especifican esto: 23672, 23673 y 23674. Si se cargan (POKE) con 0 el contador de cuadros se pone a 0. A partir de ahora se incrementan automáticamente por medio de interrupciones. He aquí un sencillo programa que utiliza esta posibilidad de hacer funcionar el ordenador como reloj o temporizador:

```
 10 POKE 23674,0: POKE
    23673,0: POKE 23672,0
 20 BORDER 0:PAPER 0:INK
    6:CLS
 30 DEF FN t()=INT
    ((65536*PEEK
    23674+256*PEEK
    23673+PEEK 23672)/50)
 40 LET t=FN t()
 50 LET h=INT (t/(60*60))
 60 LET m=INT (t/60)
 70 LET s=t-((h*60)*60)
    -(m*60)
 80 LET t$="["+STR$
    h+":"+STR$ m+":"+STR$
    s+"]"
 90 PRINT AT 1,15-((LEN
    t$)/2); "□";t$;"□"
100 GOTO 100
```

La línea 10 pone a cero el contador de cuadros y la línea 30 define la función t de lectura del contador. Esta lectura se asigna como valor de la variable t (línea 40), de donde se obtienen las horas, minutos y segundos.

EL ZOCO DE INPUT

Todo se compra y se vende. Los antiguos zocos fueron lugares destinados a todo tipo de transacciones. INPUT también tiene el suyo. Vuestras operaciones de compra, cambio o venta serán publicadas en esta sección, pero dos son las limitaciones que imponemos:

a) La propuesta tendrá que ver con la microinformática.
b) Nos reservamos el derecho de no publicar aquellos insertos de los que se sospeche un trasfondo lucrativo.

Ahora un ruego. Tratar de resumir al máximo el texto; escribir casi como un telegrama siendo claros y concisos.

Envía tu mensaje a:

INPUT SINCLAIR-ZOCO
c./ Alberto Alcocer, 46
28016 **MADRID**



EL JUEGO DE LA VIDA

- LAS REGLAS DE LA VIDA
- PATRON INICIAL
- UN CONTADOR DE GENERACIONES
- CREACION DE LA COLONIA

Cuéntale a tu microordenador unas cuantas reglas sencillas y podrás simular el comportamiento de los organismos unicelulares. El truco de este juego consiste en establecer una supervivencia óptima para la colonia.

En este juego la pantalla del televisor representa al mundo en el que pueden vivir, multiplicarse y morir las células. La pantalla queda dividida en cuadrados invisibles, cada uno de los cuales puede contener a una de las células. Si observamos el diagrama de la figura nos damos cuenta de que cada célula puede tener hasta ocho vecinos como máximo. Las reglas de la vida determinan si una célula vive o muere o, incluso, si nace una nueva.

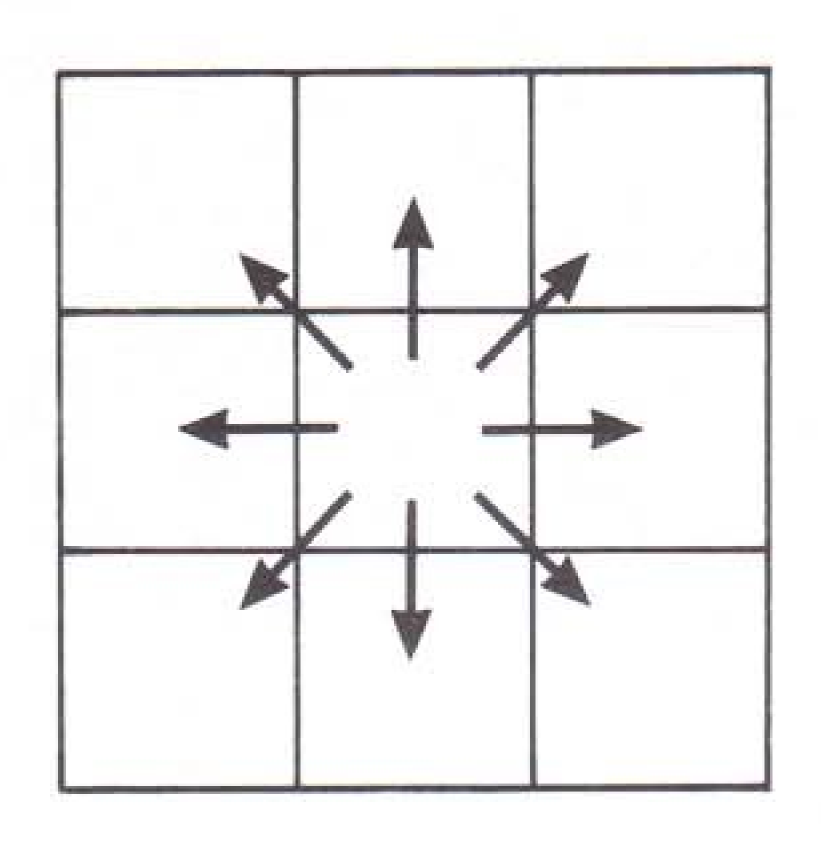
Las reglas son:

—Nacerá una célula en un espacio en el que existan exactamente tres vecinos.

—Una célula sobrevive hasta la siguiente generación si tiene dos o tres vecinos.

—El resto de las células muere.

El ordenador utiliza estas reglas para determinar el futuro de cada cuadrado de la pantalla y muestra cómo evoluciona una colonia inicial de células.



Cada célula tiene otras ocho como vecinas, de ellas depende su futuro.

Conseguir esto en BASIC sería terriblemente lento, por lo que el programa está desarrollado en código máquina, permitiendo que cada nueva generación pueda ser visualizada en aproximadamente un segundo.

El programa también muestra en la pantalla a las sucesivas generaciones en un color diferente, por lo cual el efecto resulta bastante mesmerizante cuando observamos cómo se extiende la colonia original por la superficie, dividiéndose en grupos más pequeños, muriendo o rejuveneciéndose.

La conformación de la colonia inicial es de importancia crucial, pues resulta que algunos patrones conducen a la muerte tras la existencia de sólo unas pocas generaciones, mientras que otros perduran a lo largo de centenares. Tu trabajo consiste en introducir las células que componen la colonia inicial cuyos individuos pueden reproducirse y morir.

Uno de los objetivos de este juego es diseñar un patrón de partida que sobreviva durante el mayor número de generaciones. Con este fin existe un contador de generaciones que te mantiene informado sobre lo bien que lo está haciendo tu colonia. Pero tal vez puede ser más interesante crear un colonia que se desarrolla de manera intrigante. Las fotos de pantallas que acompañan al artículo muestran unos cuantos patrones obtenidos.

CREACION DE UNA COLONIA

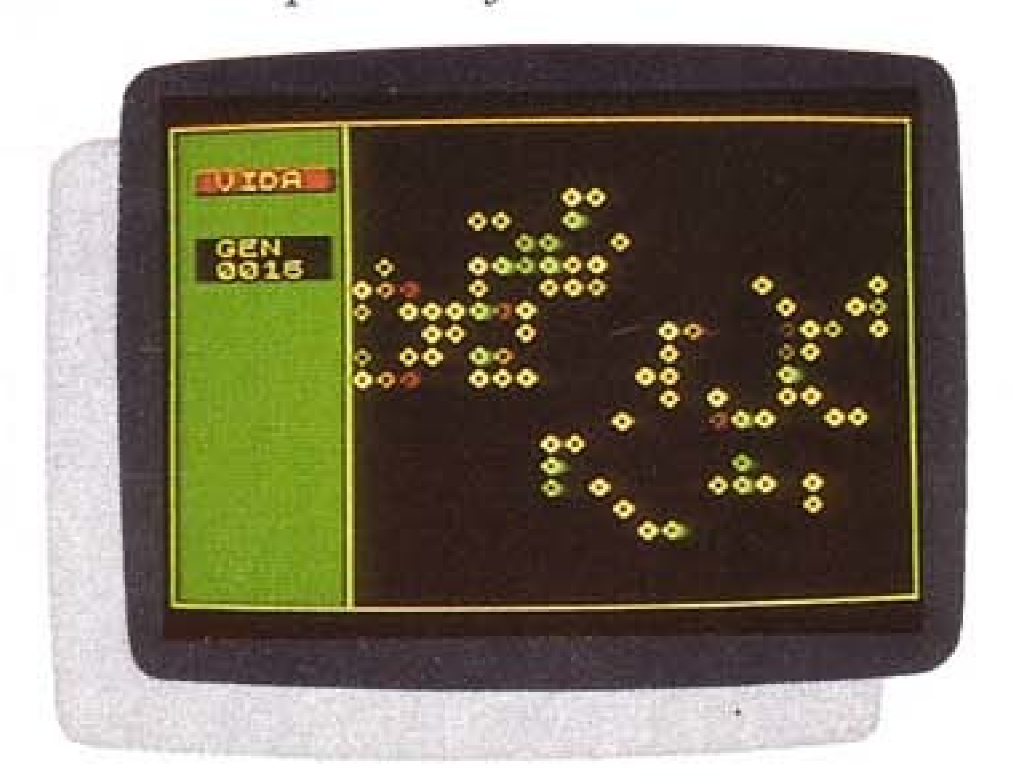
A continuación se expone cómo introducir el patrón inicial, es decir la colonia original. Para esto se utilizan las teclas de movimiento del cursor para desplazarse por toda la pantalla. ENTER sirve para crear una célula, M para borrar una célula y Q para terminar.



El programador define la colonia o conjunto de ellas a su criterio, con el fin de que permanezcan con vida el mayor número de generaciones.

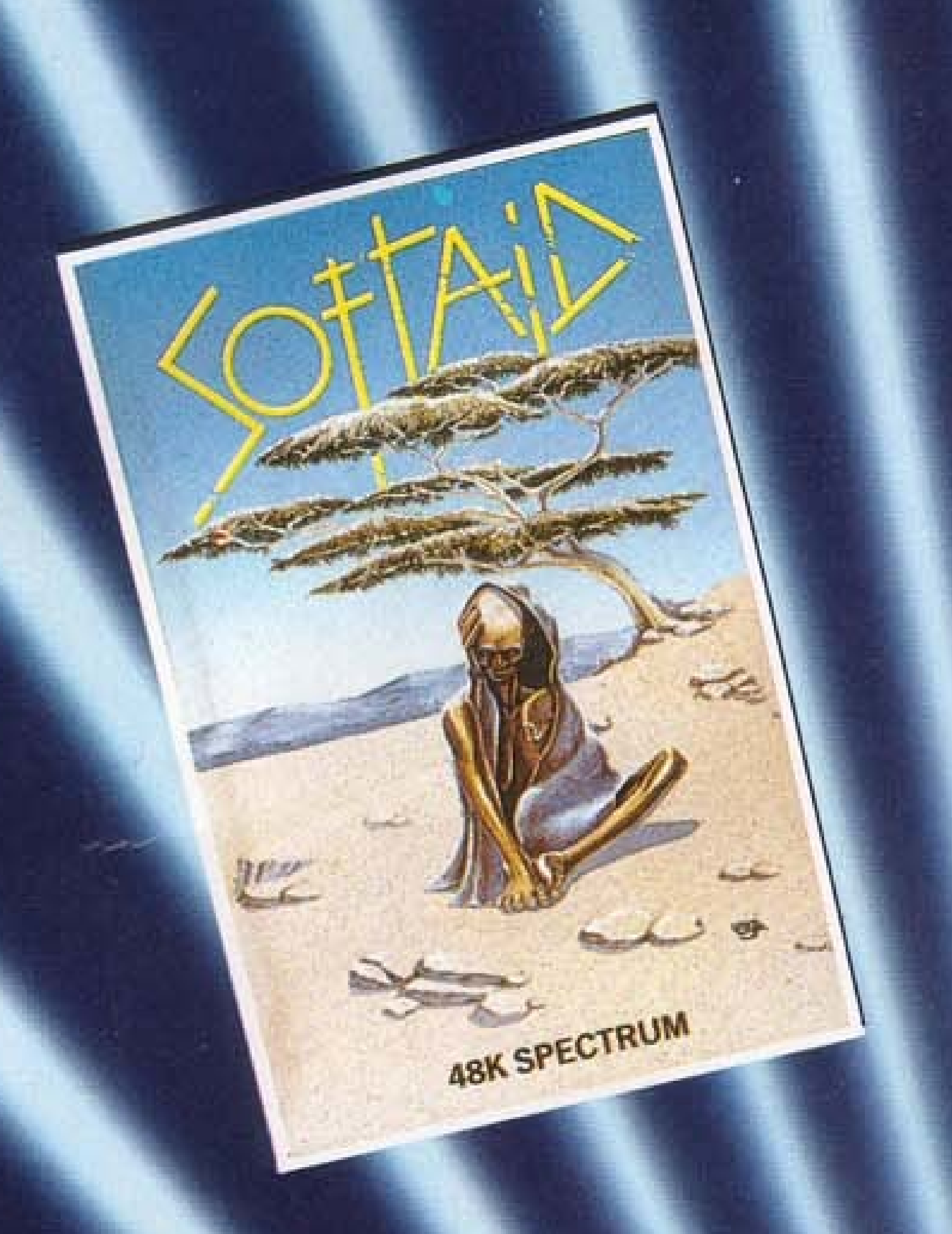


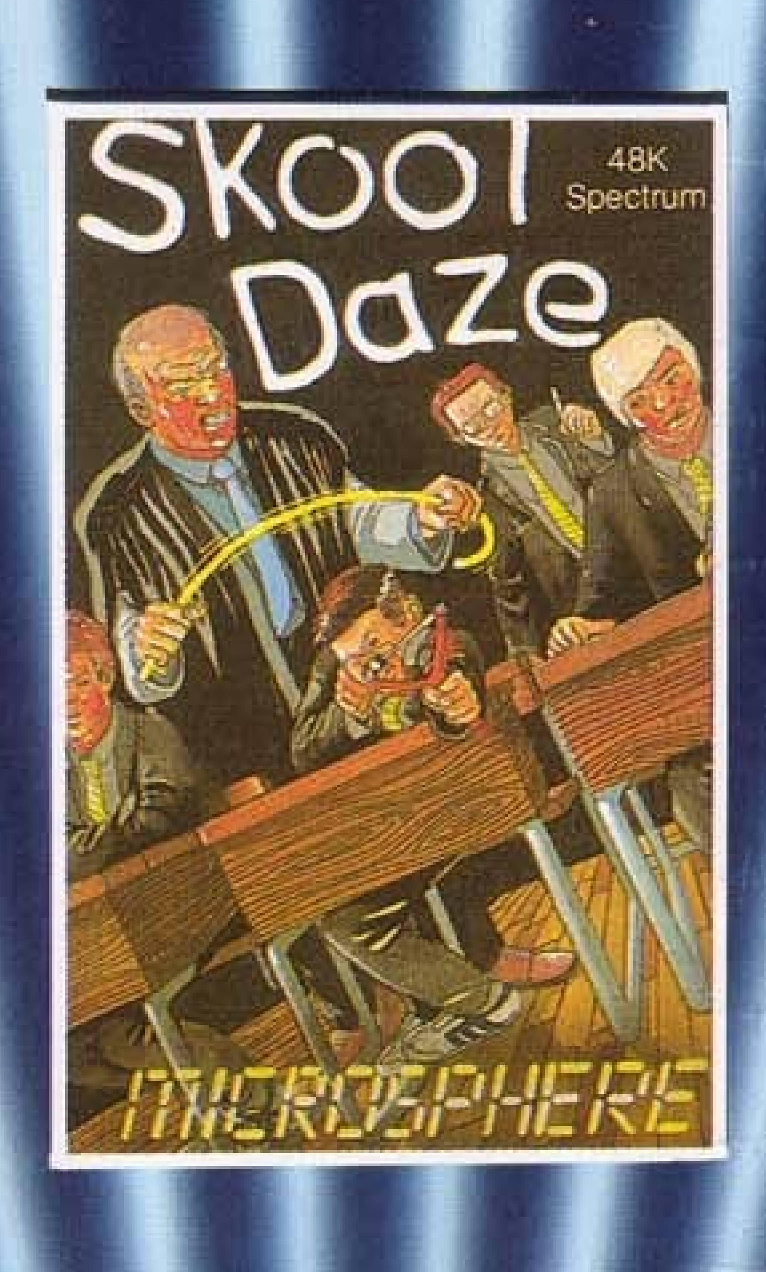
Al cabo de varias generaciones ya es factible estimar si la colonia tiene posibilidades de perdurar. En el caso de varias podrían fundirse.

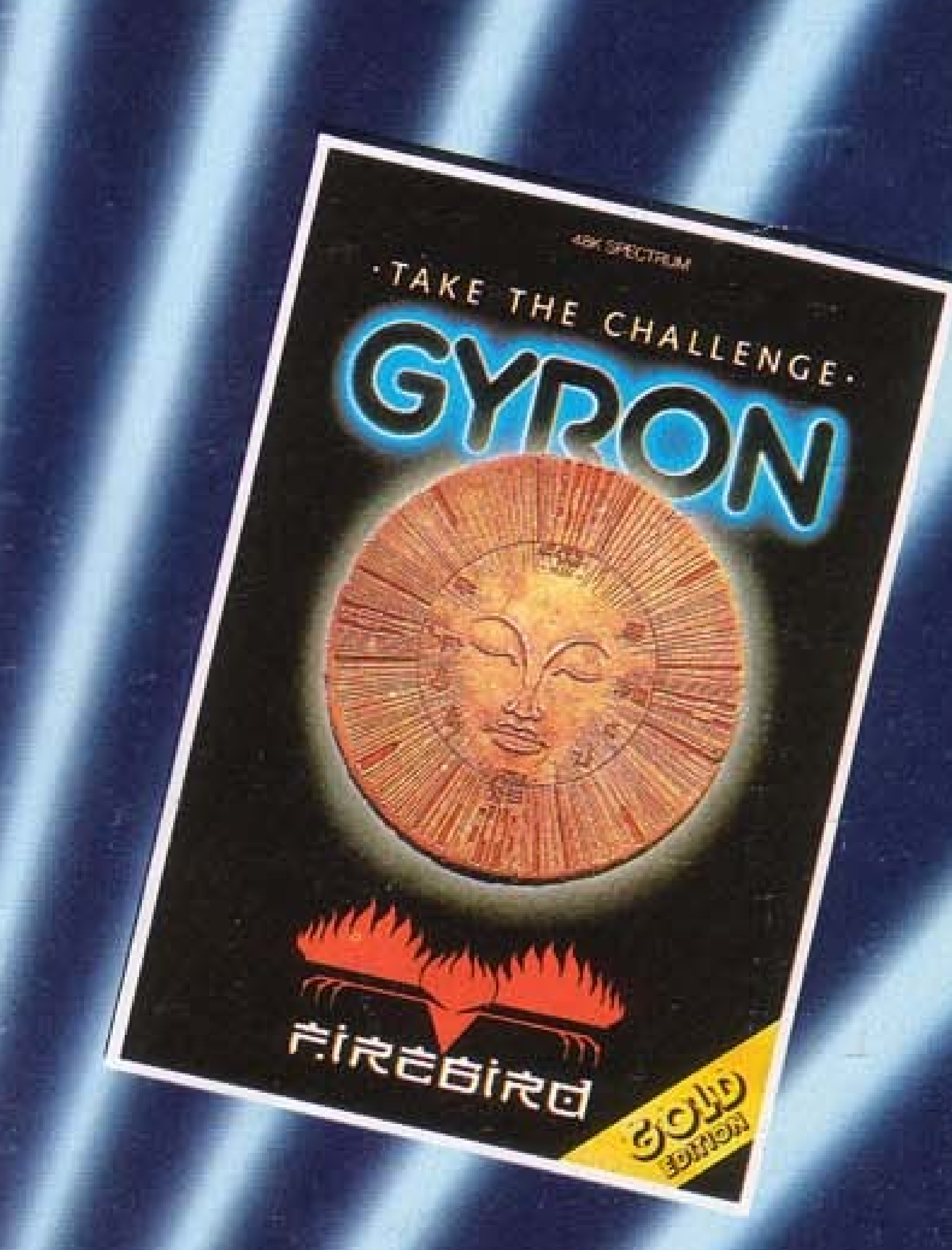


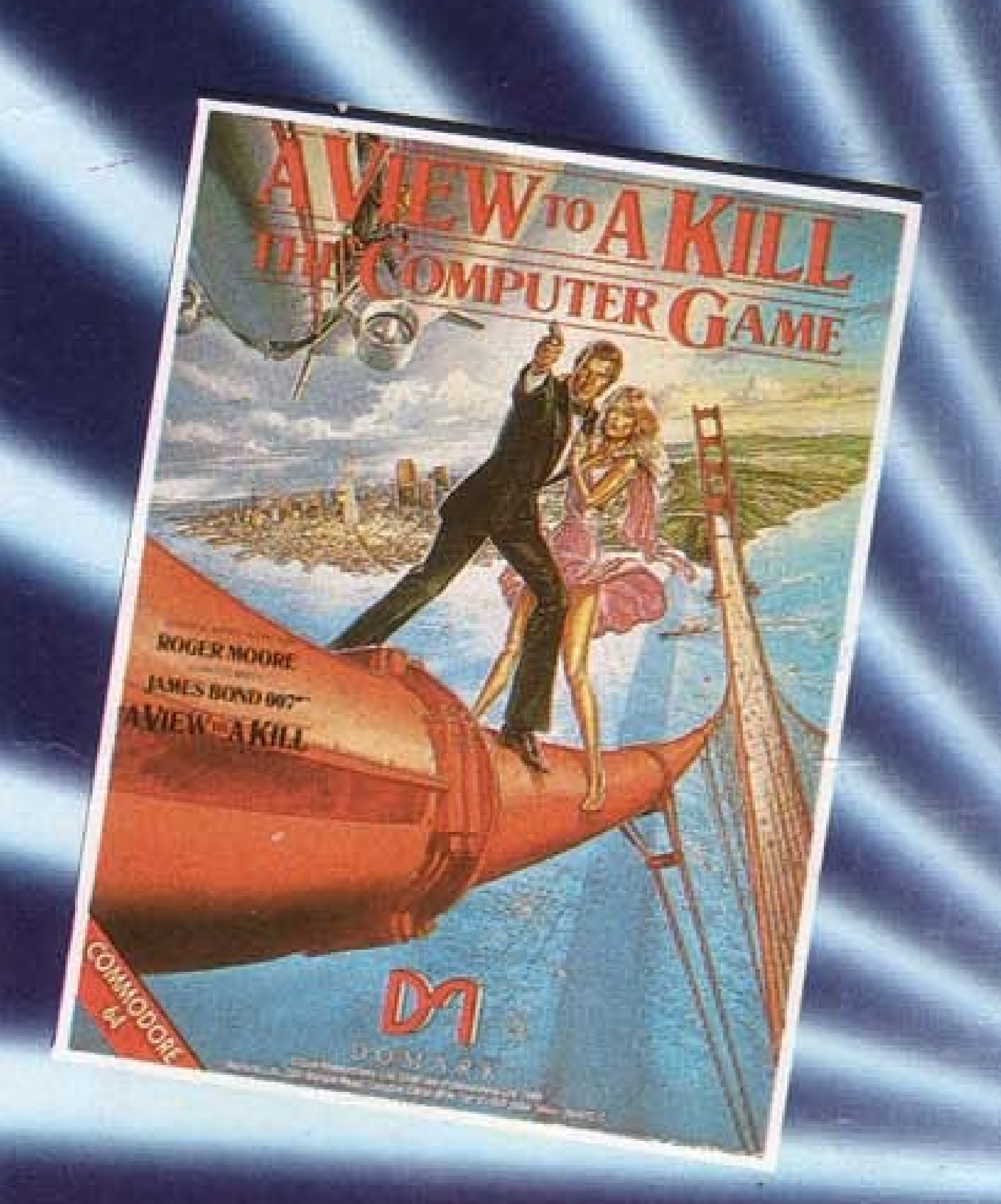
La suerte está echada. Han nacido muchas células, otras son viejas y muchas se juegan el futuro a cara o cruz.

NUESTROS EXITOS











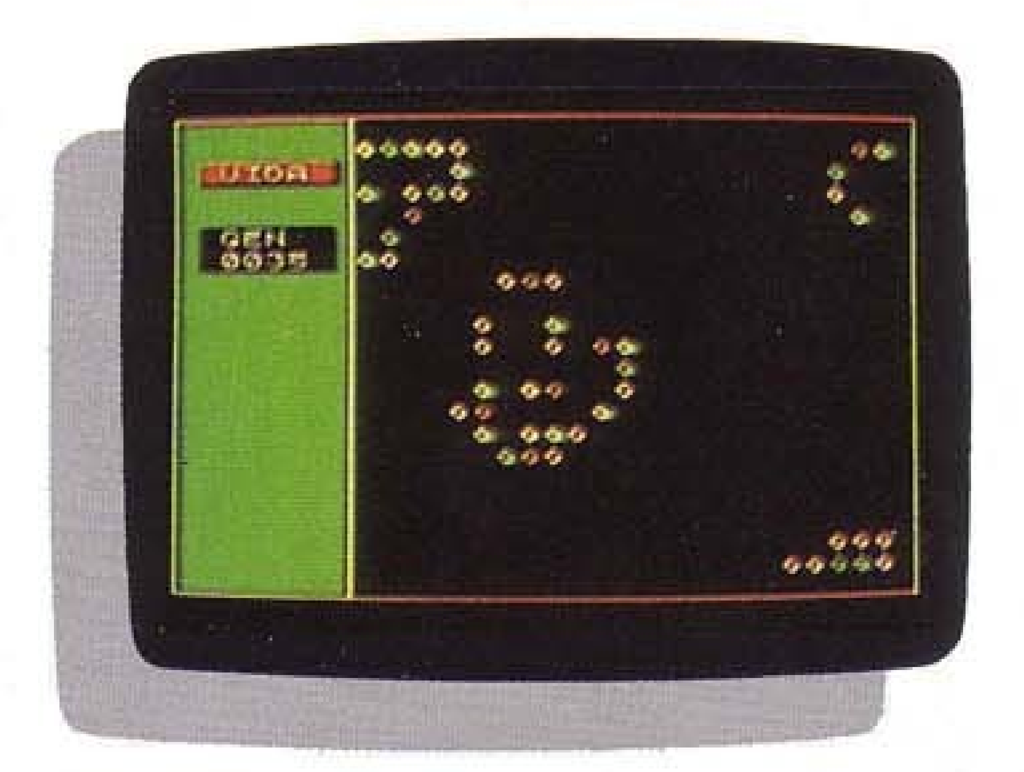




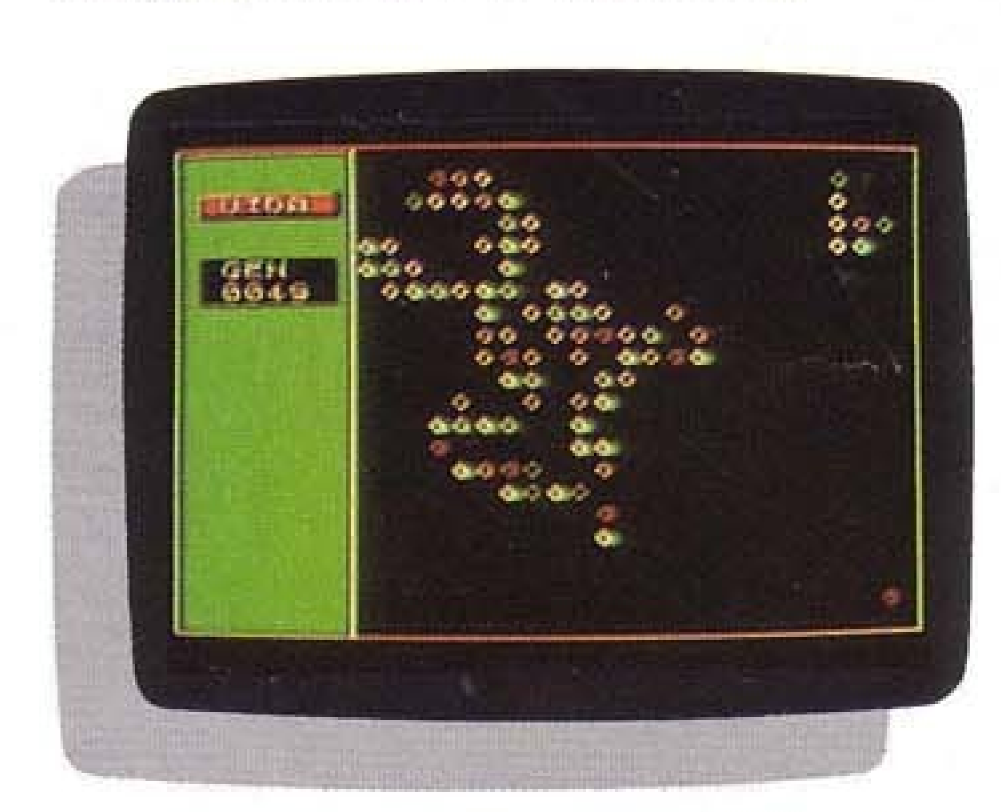
Una vez pulsada la Q el ordenador mostrará las distintas generaciones. Cuando termine podrás comenzar de nuevo pulsando cualquier tecla.

Teclea

```
5 CLEAR 28671: FOR N=USR
  "A" TO USR "A"+7: READ A:
  POKE N,A: NEXT N
6 DATA 0,24,60,102,102,
  60,24,0
7 GO SUB 200
10 POKE 23658,8: BRIGHT 1:
   BORDER 0: INK 6: PAPER 0:
   CLS
20 FOR N=0 TO 21: PRINT AT
   N,0; PAPER
   1;"□□□□□□□□": NEXT N
30 PLOT 63,0: DRAW 0,175:
   DRAW 192,0: DRAW 0,-175:
   DRAW -255,0: DRAW 0,175:
   DRAW 63,0
40 PRINT AT 2,1; PAPER 2;
   INK 7;"□VIDA□"; PAPER
   0; INK 6; AT
   5,1;"□GEN□"; AT
   6,1;"□0000□"
70 RANDOMIZE USR 28672: LET
   X=20: LET Y=10
80 PRINT AT Y,X; OVER
   1;"■": FOR N=1 TO 10:
   NEXT N: PRINT AT Y,X;
   OVER 1;"■"
90 LET A$=INKEY$: IF A$=""
   THEN GO TO 80
92 IF A$="Q" THEN GO TO 110
94 IF CODE A$=13 THEN PRINT
   AT Y,X;CHR$ 144: POKE
   30000+((Y-1)*23)+(X-8),1
   44: GO TO 80
95 IF A$="M" THEN PRINT AT
   Y,X;"□": POKE
   30000+((Y-1)*23)+(X-8),3
   2: GO TO 80
100 LET X=X-(A$="5")*
    (X>8)+(A$="8")*(X<30):
    LET Y=Y-(A$="7")*(Y>1)
    +(A$="6")*(Y<20): GO
    TO 80
110 OVER 0: RANDOMIZE USR
    28711
120 PRINT AT 21,7; FLASH
    1;"UNA TECLA PARA
    RECOMENZAR": FOR N=1 TO
    200: NEXT N
130 LET A$=INKEY$: IF A$=""
    THEN GO TO 130
140 GO TO 10
200 LET L=500: RESTORE L:
    FOR N=28672 TO 28951
    STEP 8
210 LET T=0: FOR D=0 TO 7
220 READ A: POKE N+D,A: LET
    T=T+A: NEXT D: READ A: IF
    A<>T THEN PRINT FLASH
    1;"ERROR EN EL DATA DE LA
    LINEA□";L: STOP
230 LET L=L+10
240 NEXT N
250 RETURN
500 DATA 33,25,117,1,211,3,
    62,32,484
510 DATA 119,35,13,32,249,
    5,242,6,701
520 DATA 112,33,48,48,34,0,
    113,34,422
530 DATA 2,113,33,48,117,
    34,252,112,711
540 DATA 33,248,118,34,254,
    112,201,243,1243
550 DATA 42,252,112,6,1,
    197,88,6,704
560 DATA 8,80,62,22,215,
    123,215,122,847
570 DATA 215,126,35,254,32,
    40,10,214,926
580 DATA 142,245,62,16,215,
    241,215,62,1198
590 DATA 144,215,4,120,254,
    31,32,233,1033
600 DATA 193,4,120,254,21,
    32,214,42,880
610 DATA 252,112,237,91,
    254,112,229,213,
    1500
620 DATA 1,200,1,197,221,
    33,4,113,770
630 DATA 1,0,7,213,221,94,
    0,221,757
640 DATA 86,1,229,25,235,
    225,26,254,1081
650 DATA 32,209,40,1,12,
    221,35,221,771
660 DATA 35,5,242,107,112,
    126,254,144,1025
670 DATA 121,56,18,254,2,
    40,4,254,749
680 DATA 3,32,14,126,254,
    32,32,11,504
690 DATA 58,251,112,24,6,
    254,3,40,748
700 DATA 242,62,32,18,35,
    19,193,13,614
```



Las generaciones se suceden…



…pero el juego nunca pierde interés.

```
710 DATA 32,185,5,242,99,
    112,209,225,1109
720 DATA 237,83,252,112,34,
    254,112,33,1117
730 DATA 4,113,43,126,60,
    254,58,40,698
740 DATA 4,119,195,203,112,
    62,48,119,862
750 DATA 195,186,112,62,22,
    215,62,6,860
760 DATA 215,62,2,215,33,0,
    113,62,702
770 DATA 16,215,62,6,215,6,
    4,126,650
780 DATA 35,215,16,251,58,
    251,112,60,998
790 DATA 254,151,32,2,62,
    144,50,251,946
800 DATA 112,62,127,219,
    254,31,218,40,1063
810 DATA 112,251,201,149,
    248,118,48,117,1244
820 DATA 48,49,54,49,233,
    255,234,255,1177
830 DATA 1,0,24,0,23,0,22,
    0,70
840 DATA 255,255,232,255,0,
    0,0,0,997
```

INPUT JUEGOS

COLECCIONABLE DE PROGRAMACION



Aquí aprenderás cómo diseñar tus propios juegos para ordenador; desde los más simples a los más complejos de estrategia e inteligencia. Una gran abundancia de ejemplos son la manera más agradable de introducirte en los secretos de la programación.

MOVIMIENTO Y ANIMACION

- LOS PRINCIPIOS DE LA ANIMACION
- MOVIMIENTO DE GRAFICOS
- COMO UTILIZAR LOS GRAFICOS INCORPORADOS

¿Quieres darle vida a tus juegos de programación? Empieza entonces con estos sencillos caracteres gráficos, que puedes generar a partir de la memoria gráfica de tu ordenador.

Jugar con los juegos que se venden para tu ordenador es divertido sólo hasta cierto punto, y llega un momento en que la mayor parte de la gente siente la necesidad de dar rienda suelta a su imaginación y crear programas de juegos propios.

La programación de juegos no es fácil; tienes que empezar con cosas muy sencillas e ir aumentando poco a poco en complejidad. Pero eso te ayudará a pensar con lógica y aumentará tu habilidad como programador. Y también te divertiras más.

Lo primero que tienes que aprender en la programación de juegos, aparte de los trucos del BASIC, es la técnica de la animación.

Para crear la ilusión de movimiento, el programador de ordenadores utiliza en gran parte la misma técnica que el caricaturista que anima una película de dibujos. Lo que hace es crear dos (o más) imágenes y alternarlas rápidamente (idealmente, unas 24 veces por segundo).

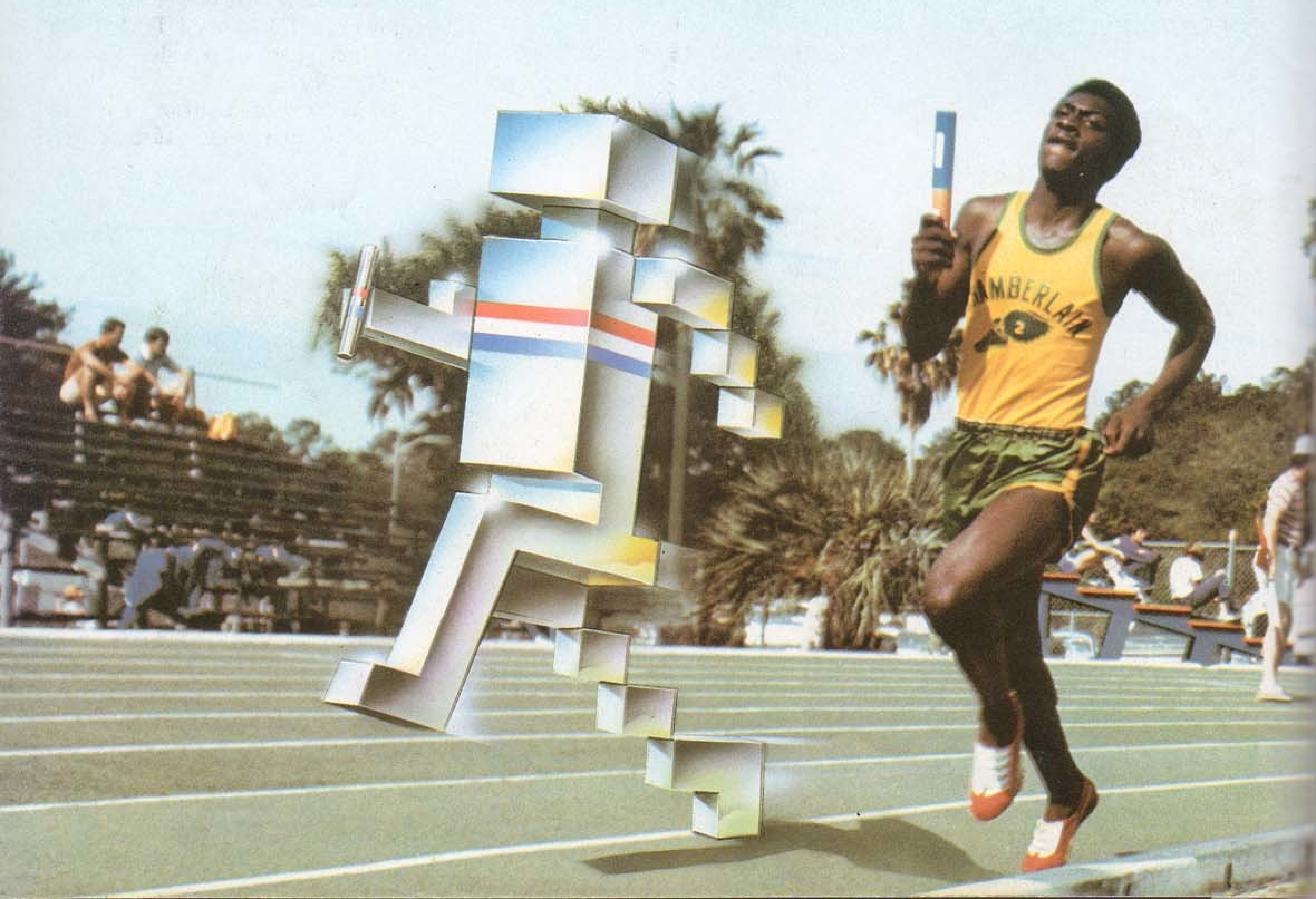
Pero existe una diferencia importante. En la animación de dibujos, el dibujante cuenta con el proyector de películas, que le permite olvidarse de una imagen cuando ya no la necesita. En la animación por medio de un ordenador no ocurre esto. Como no hagas algo para evitarlo, cualquier segmento de imagen que "proyectes" sobre un área dada de la pantalla permanecerá allí indefinidamente.

Una forma de olvidarse de la imagen que ya no se necesita es, simplemente, imprimir algo sobre ella. El ordenador no puede situar dos imágenes al mismo tiempo en la misma posición de la pantalla.

Así si, por ejemplo, la línea 10 de un programa cualquiera dice al ordenador que imprima una A en una determinada posición, cualquier otra línea de programa que imprima (por ejemplo) una B en la misma posición se desembarazará de la A.

Los programas que siguen contienen varios ejemplos de esta clase de sustitución.

¿Pero qué ocurre si no tienes nada



que imprimir encima del carácter no deseado? Acuérdate en ese caso de incluir en alguna línea posterior una instrucción que imprima un espacio en la correspondiente posición de la pantalla.

Si te olvidas de este detalle, tu pantalla pronto se verá abarrotada con trozos no deseados de brazos, piernas y cuerpos.

La manera de obtener los caracteres gráficos en la pantalla difiere mucho de un ordenador a otro. Hay diferencias entre los caracteres gráficos incorporados en ROM y lo mismo ocurre con la forma en que se imprimen (con PRINT) en pantalla. Por último también hay diferencias en el modo de hacer que se muevan.

Más abajo se muestra un pequeño ciempiés cuya animación resulta muy fácil con el **Spectrum** o **ZX81.** Para empezar, intenta crearlo en una posición estática, así:

```
)))     )))     )))     )))
OOO<    OOO<    OOO<    OOO<
)))     )))     )))     )))
```

```
10 PRINT AT 10,15;"000"
20 PRINT AT 9,15;")))"
30 PRINT AT 11,15;")))"
40 PRINT AT 10,18;"<"
```

Ejecuta ahora el programa (RUN). Aunque esta es una manera bastante elaborada de generar una imagen sencilla, sirve para ilustrar uno o dos puntos interesantes.

En primer lugar, te da una idea de las posiciones relativas dentro de la pantalla. El centro del insecto está en la línea 10, empezando a contar por arriba, aproximadamente en la mitad de la pantalla.

En segundo lugar, muestra el efecto que resulta de decir al ordenador que ponga más de un carácter al mismo tiempo en una misma posición: Lo que hace es simplemente colocar los caracteres extra en las posiciones contiguas. Por eso los "palpadores" de la línea 40 de programa están en 10, 18. Las posiciones 10, 15; 10, 16, y 10, 17; ya estaban ocupadas por el cuerpo del insecto.

Una forma más adecuada de generar el insecto en pantalla es reunir las instrucciones anteriores en una sola línea de programa, así:

```
10 PRINT AT 10,15;"000";AT
   9,15;")))";AT 11,15;")))"
```

Añádele estas dos líneas y tendrás ya animación:

```
20 PRINT AT 10,15;"000<";AT
   9,15;"(((";AT 11,15;"((("
30 GOTO 10
```

Cuando ejecutes (RUN) este programa, verás que se produce un emborronamiento, se debe a que se cambia de imagen demasiado rápidamente.

Para que vayan más despacio, lo más cómodo es poner un bucle FOR-...NEXT, que hace que el ordenador cuente hasta 10 (o el número que tú pongas) antes de presentar en pantalla la siguiente imagen. Añade al programa estas líneas suplementarias. Para el **ZX81** utiliza 2 en vez de 10.

```
15 FOR L=1 TO 10
17 NEXT L
25 FOR M=1 TO 10
27 NEXT M
```

Naturalmente, puedes modificar la duración de la pausa sin más que cambiar el 1 TO 10 por 1 TO 5, 1 TO 20, o cualquier otro valor que te resulte mejor.

Ya has conseguido crear un pequeño insecto más bien inútil, cuyas



patas se sacuden desesperadamente, pero que no se mueve. Eso vendrá un poco después (*véase* Movimiento).

```
)))     )))     )))     )))
OOO<    OOO<    OOO<    OOO<
)))     )))     )))     )))
```

USO DE LOS GRAFICOS DE LA ROM

Se puede producir una animación más interesante con los caracteres gráficos estándar de **Sinclair.** En la figura 3 tienes un ejemplo.

Más adelante tienes el programa completo, pero si no estás acostumbrado a usar los símbolos gráficos, será útil que te entretengas primero en crear una figura estática línea por línea, así:

```
1 PRINT AT 5,15;"O"
2 PRINT AT 6,14;"▞█▚"
```

...y así sucesivamente.

Es fácil obtener estos caracteres gráficos. Por ejemplo, para la línea 2 anterior, pulsa al mismo tiempo [CAPS SHIFT] ([SHIFT] en el caso del **ZX81**) y 9. Esto te sitúa en el modo gráfico, indicado con una G intermitente en la pantalla. Ahora pulsa, en el **Spectrum,** [CAPS SHIFT] y 6 juntos para tener la versión en video inverso (negro sobre blanco) del símbolo correspondiente a la tecla 6; a continuación pulsa [CAPS SHIFT] y 8; después el propio 6. En el **ZX81** pulsa [SHIFT] y T; [SHIFT] y [SPACE]; [SHIFT] e Y. Finalmente, pulsa otra vez 9 para salir del modo gráfico, antes de insertar las comillas al final.

Aquí tienes todo el programa de la figura:

```
1Ø PRINT AT 5,14;"□O□";AT
   6,14;"▞█▚";AT
   7,14;"▗█▖";AT
   8,14;"▟□▀"
2Ø PRINT AT 5,14;"□O□";AT
   6,14;"▞█▚";AT
   7,14;"▚█▖";AT
   8,14;"▖□▙"
3Ø GOTO 1Ø
```

También aquí verás que para ralentizar la acción es preciso insertar un bucle FOR...NEXT después de la línea 10 y otro después de la línea 20.



MOVIMIENTO

Aquí tienes, por fin, un programa que no sólo anima al insecto del programa anterior, sino que también hace que se mueva por la pantalla:

```
1Ø FOR N=Ø TO 27
2Ø PRINT AT 1Ø,N;"OOO<";AT
   9,N;"(((";AT 11,N;"((("
3Ø PRINT AT 1Ø,N;"□□□□";AT
   9,N;"□□□";AT
   11,N;"□□□"
4Ø PRINT AT 1Ø,N;OOO<";AT
   9,N;")))"AT 11,N;")))"
5Ø PRINT AT 1Ø,N;"□□□□"AT
   9,N;"□□□";AT
   11,N;"□□□"
6Ø NEXT N
7Ø GO TO 1Ø
```

Este programa también hace uso de un bucle FOR...NEXT, pero con una finalidad completamente distinta. En el ejemplo anterior, el ordenador estaba contando una fracción de segundo antes de presentar una imagen. Ahora lo que hace es mover la imagen por la pantalla, una posición de pantalla (o si quieres una "casilla") cada vez.

¿Por qué se lee entonces en la línea 10 "0 TO 27", cuando hay de hecho 32 posiciones de pantalla en la pantalla de un **Sinclair**? Para ver el porqué, prueba a sustituir la línea 10 del programa por esta otra:

```
1Ø FOR N=Ø TO 32
```

Posiblemente también te haya extrañado la presencia de las líneas 30 y 50. Pero suprímelas y en seguida te darás cuenta de por qué hacen falta.

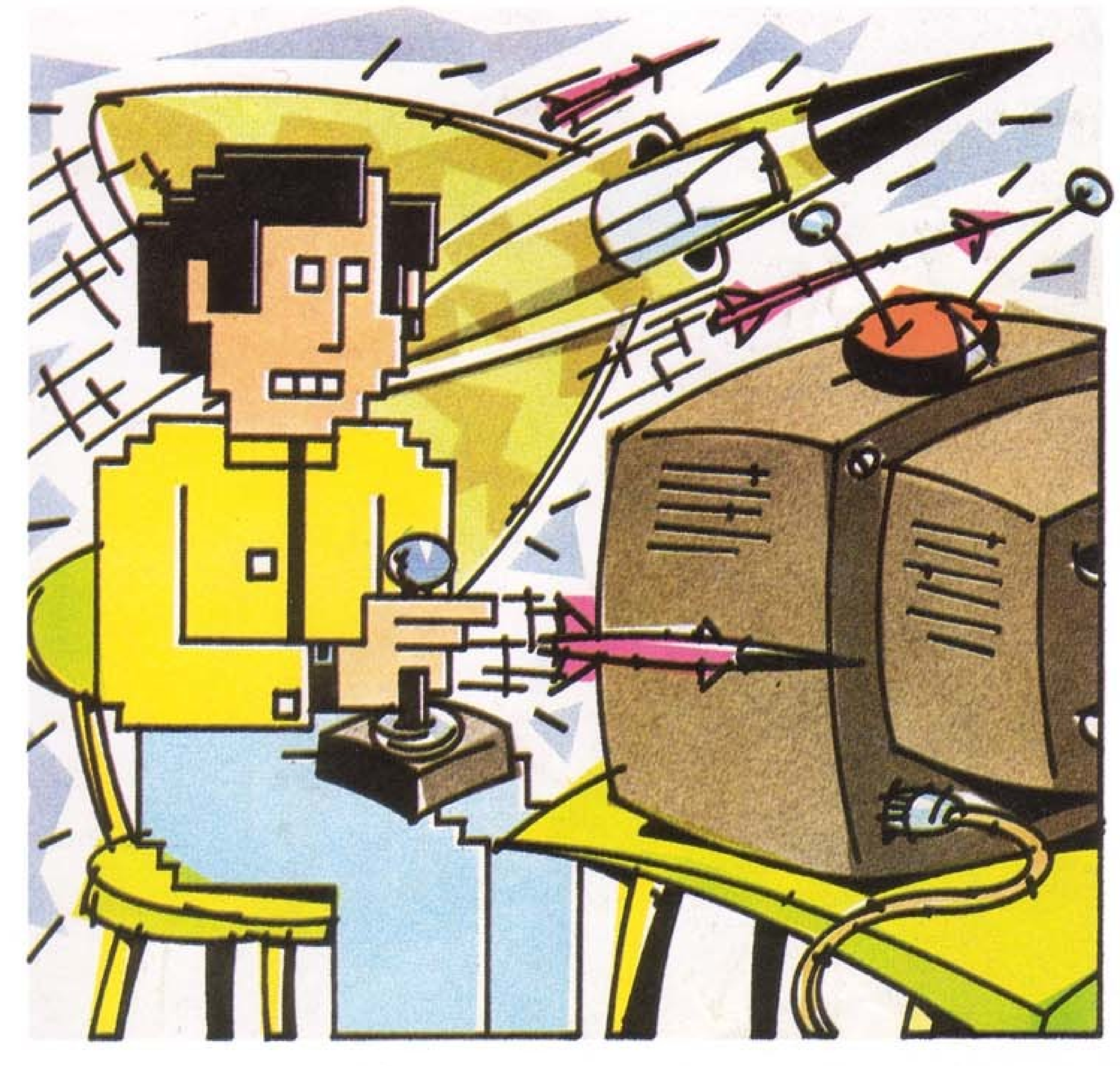
LOS SIGUIENTES PASOS

Ahora que has empezado a aprender la animación y el movimiento, puedes inventar tus propias figuras, cohetes, o naves espaciales, utilizando los gráficos que vienen en tu manual, los caracteres de máquina de escribir, o ambas cosas.

DERECHA... IZQUIERDA... ARRIBA... ¡FUEGO!

- DETECCION DE LAS PULSACIONES DE TECLA
- LANZAMIENTO DE MISILES
- CONTROL DE UN GRAFICO MOVIL

Los juegos de marcianitos se basan en la habilidad del jugador para controlar los sucesos de la pantalla. He aquí cómo controlar el movimiento, lanzar misiles e integrarlos en un juego.



Los juegos de marcianitos serían tremendamente aburridos si el movimiento básico del láser o los disparos no pudieran ser controlados de alguna manera. Este tipo de control desde el teclado no es más que una faceta de los juegos de invasores, incluso de los más simples. Por eso es importante conocer los principios si pretendes escribir programas propios.

El primer paso se centra en conseguir que el ordenador reaccione cuando presiones una tecla.

DETECCION DE UNA TECLA PULSADA

En principio, todos los microordenadores emplean el mismo método para detectar una pulsación de tecla, aunque en detalle existan grandes variaciones.

Cuando el ordenador encuentra la función INKEY$ en un programa, explora el teclado para ver si ha sido presionada una tecla. A continuación vemos un breve programa que utiliza INKEY$:

```
20 CLS
30 IF INKEY$="" THEN GO TO 30
40 PRINT AT 11,14;"AUUG"
```

Ejecuta el programa y después presiona cualquier tecla, excepto CAPS SHIFT o SYMBOL SHIFT. La máquina visualiza "AUUG". El programa trabaja de la siguiente forma:

La línea 20 limpia la pantalla. La línea 30 hace que el ordenador espere hasta que sea presionada una tecla antes de continuar con el programa. Date cuenta que no está incluido un espacio entre las comillas. Por ello, la línea 30 significa: "Si INKEY-$=nada, o si no ha sido presionada tecla alguna, vuelve a comprobar." Es importante disponer del IF... THEN GOTO 30, porque de otra forma el ordenador comprobaría solamente una vez si ha sido presionada una tecla y sólo durante una fracción de segundo.

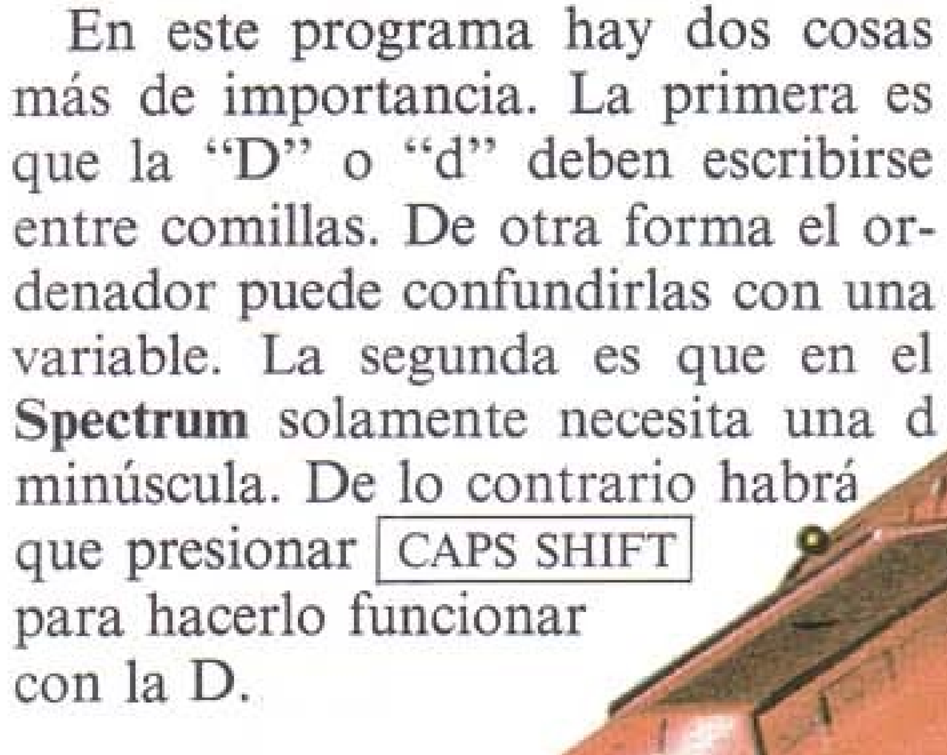
Tan pronto como la tecla sea pulsada, INKEY$ toma el valor de esa tecla. Por ejemplo, si presionamos el 3, entonces INKEY$="3". Esto es suficiente para que la línea 40 visualice "AUUG" en la pantalla.

En la mayoría de los juegos debes presionar una determinada tecla para mover un tanque, una nave espacial o cualquier otra cosa. Si alteras la línea 40 verás cómo se consigue esto. En el **ZX81,** utiliza la D mayúscula, borra el :STOP y añade 45 STOP:

```
40 IF INKEY$="d" THEN PRINT
   AT 11,9;"ESO ESTA BIEN":
   STOP
50 PRINT AT 11,14;"AUUG!!"
```

La línea 40 comprueba si la tecla D ha sido presionada; en otras palabras ¿Es INKEY$ igual a D o d? Si no es así, el **Spectrum,** que no dispone de la sentencia ELSE como otros ordenadores, ignorará la línea 40 y continuará en la 50 (trata de descubrir por qué es necesario el STOP).

En este programa hay dos cosas más de importancia. La primera es que la "D" o "d" deben escribirse entre comillas. De otra forma el ordenador puede confundirlas con una variable. La segunda es que en el **Spectrum** solamente necesita una d minúscula. De lo contrario habrá que presionar CAPS SHIFT para hacerlo funcionar con la D.





El programa incluido a continuación dispara un misil cuando se presiona la f. En el **ZX81** se utilizan mayúsculas y el asterisco en lugar de la flecha de la línea 60.

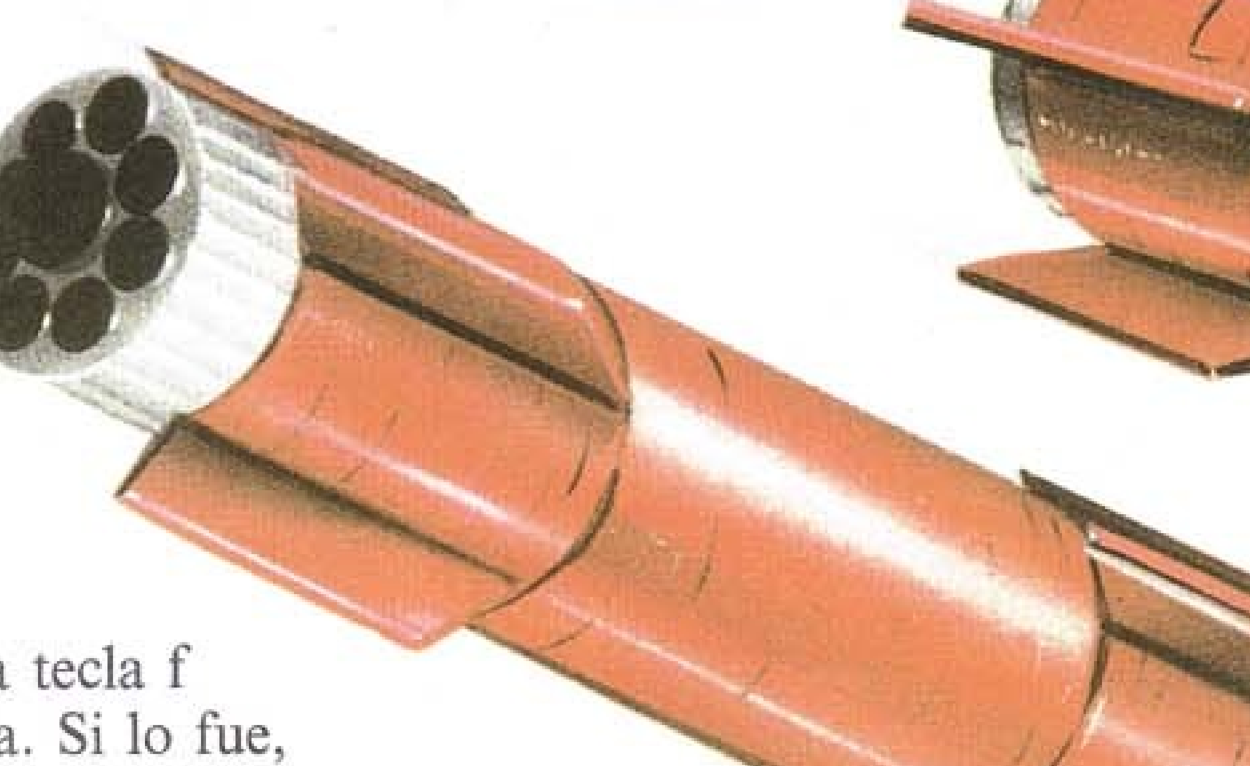
```
20 CLS
30 PRINT AT 21,14;"▄█▄"
40 IF INKEY$="" THEN GO TO 40
50 IF INKEY$<>"f" THEN GO TO
   40
55 LET y=20
60 PRINT AT y,15;"↑"
70 LET y=y-1
75 PAUSE 1
80 PRINT AT y+1,15;"□"
90 IF y>0 THEN GO TO 60
```

La línea 30 utiliza los gráficos de baja resolución de la ROM.

La línea 40, como antes, hace que el **Spectrum** espere hasta la presión de una tecla.







La línea 50 comprueba si la tecla f ha sido activada. Si lo fue, queda establecida la posición de comienzo del misil. Esto es en la vigésima línea de la pantalla, una por encima de la base del misil.

La línea 40 hace que el ordenador explore el teclado de nuevo si es presionada otra tecla distinta a la f.

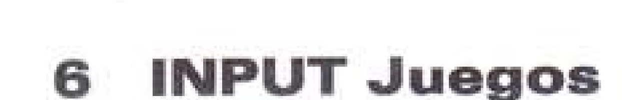
En la línea 60 se visualiza el misil en la pantalla. El carácter en el **Spectrum** es una flecha apuntando hacia arriba, que se obtiene presionando SYMBOL SHIFT y H.

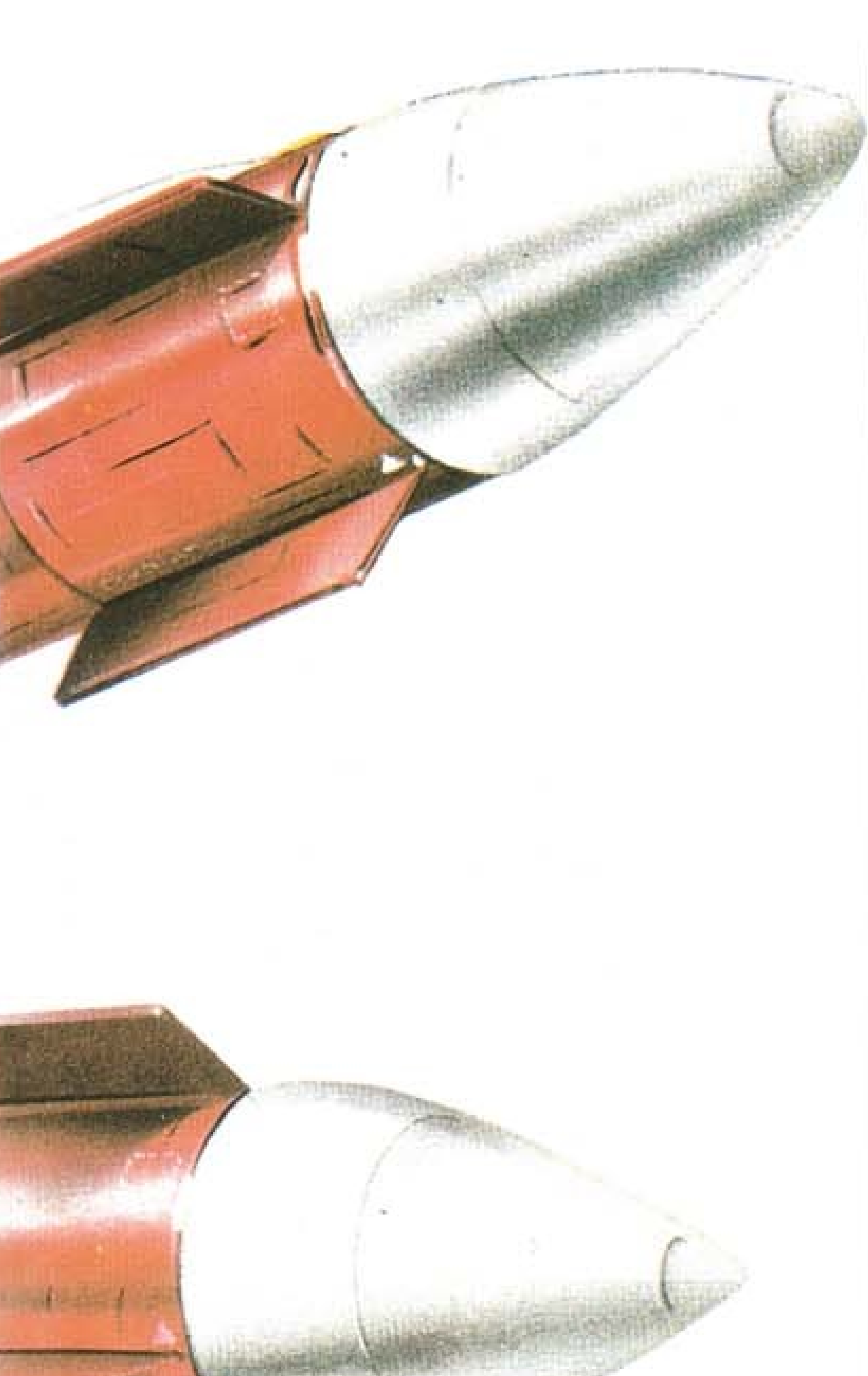
La línea 80 borra la posición previa del misil.

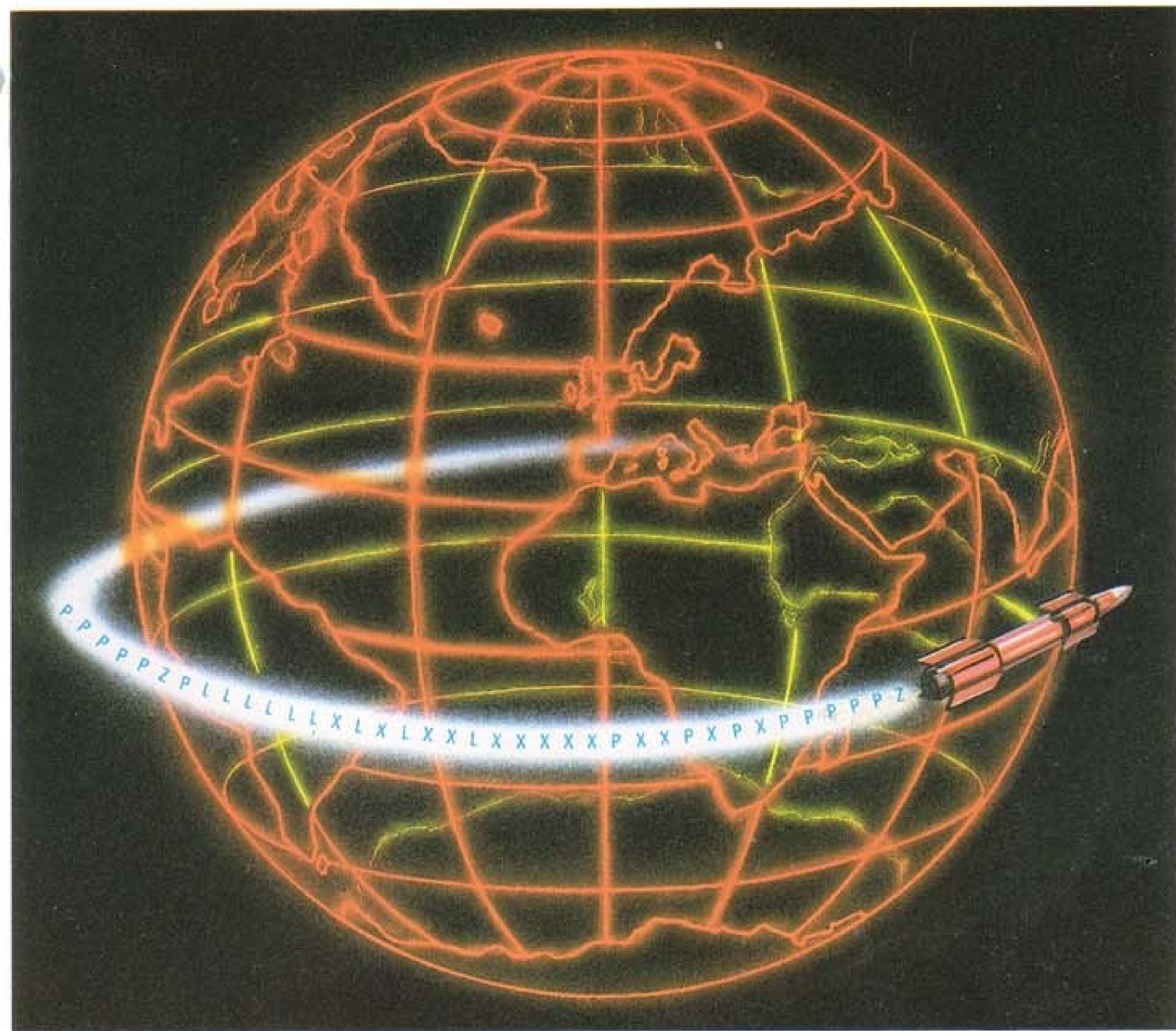
La línea 70 sustrae 1 de y, que es la línea de coordenada de la posición del misil, desplazándole así una posición más arriba en la pantalla.

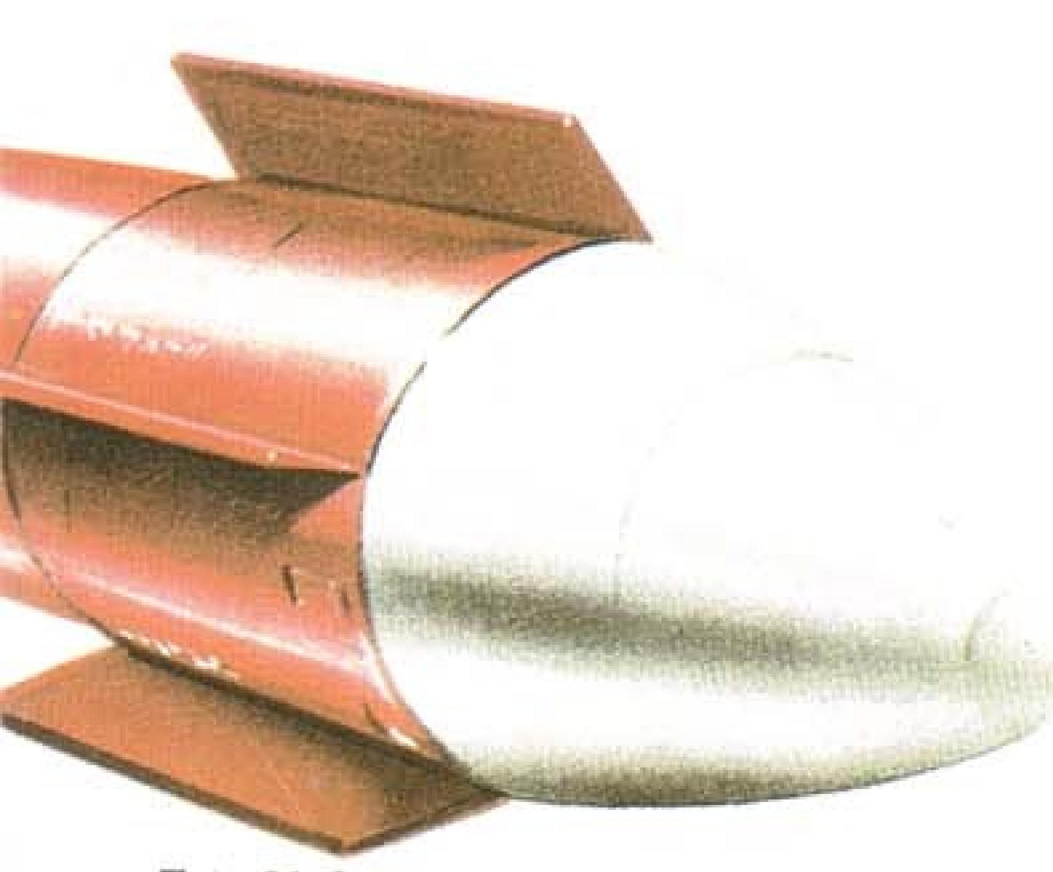
La línea 90 detiene la salida del misil fuera de la pantalla (cuando alcanza la línea superior, la coordenada y es 0).

El siguiente programa desplaza la base del misil por la pantalla. En el **ZX81** se tecleará todo el programa empleando las mayúsculas.









```
30 CLS
40 LET x=15
50 LET y=13
60 PRINT AT y,x;"▄█▄"
70 IF INKEY$="" THEN GO TO
   70
80 LET lx=x: LET ly=y
90 PRINT AT ly,lx;"□□□"
100 IF INKEY$="f" THEN STOP
110 IF INKEY$="p" THEN LET
    y=y-1
120 IF INKEY$="l" THEN LET
    y=y+1
130 IF INKEY$="z" THEN LET
    x=x-1
140 IF INKEY$="x" THEN LET
    x=x+1
150 IF x<1 OR x>29 THEN LET
    x=lx
160 IF y<1 OR y>20 THEN LET
    y=ly
170 GO TO 60
```

Las líneas 40 y 50 ajustan la posición inicial de la base del misil, 13 líneas a partir de arriba y 15 espacios desde la izquierda. La línea 60 visualiza la base en la pantalla.

La línea 70 hace esperar al **Spectrum** hasta que se presione una tecla.

Las líneas 80 y 90 son tal vez las de más difícil comprensión. Pero lo que consiguen es hacer que una hilera de tres espacios siga a la base del misil por la pantalla. Debido a que la línea 60 viene en el bucle siempre antes que la 90, la base en su nueva posicion es dibujada (mediante PRINT) antes de que sea borrada en su antigua posición.

La línea 100 acaba el programa si presionamos la f (fin).

La línea 110 comprueba si se ha presionado p, y si lo ha sido, resta 1 al valor de y. El efecto que tiene esto es desplazar la base un "espacio" hacia arriba.

Las líneas 120 a 140 operan de modo similar. La 120 desplaza la base hacia abajo si presionamos l; la 130 hacia la izquierda con la z, y la 140 produce el desplazamiento a la derecha si optamos por la x.

La línea 150 previene que la base salga fuera de los márgenes de la pantalla. La línea 160 pretende lo mismo pero con los márgenes superior e inferior.

La línea 170 completa el bucle, haciendo que sea explorado el teclado y recomenzando el proceso.

CREACION DE UN JUEGO

Ahora que ya dispones de algunos bloques puedes construir juegos a partir de ellos. El presentado más abajo muestra una manera de utilizarlos.

En el **ZX81** se teclea en mayúsculas y se utiliza el asterisco en lugar de la flecha. En la línea 40 se borra LET y=21 y se añade 45 LET Y=21:

```
 20 CLS
 30 PAUSE 25
 40 LET x=15: LET y=21
 50 LET B$="□▄█▄⊏"
 60 LET a=INT (RND*28)+2
 70 PRINT AT 2,a;"*"
 80 LET xx=x
 90 PRINT AT y,x;B$
100 IF INKEY$="z" THEN LET
    x=x-1
110 IF INKEY$="x" THEN LET
    x=x+1
120 IF x<0 OR x>27 THEN LET
    xx=x
140 IF INKEY$<>"f" THEN GO
    TO 80
145 LET m=y-1
150 PRINT AT m,x+2;"⇧"
160 PRINT AT m+1,x+2;"□"
170 LET m=m-1
180 IF m=2 AND x+2=a THEN GO
    TO 20
190 IF m<>1 THEN GO TO 150
195 PRINT AT m+1,x+2;"□"
200 GO TO 80
```

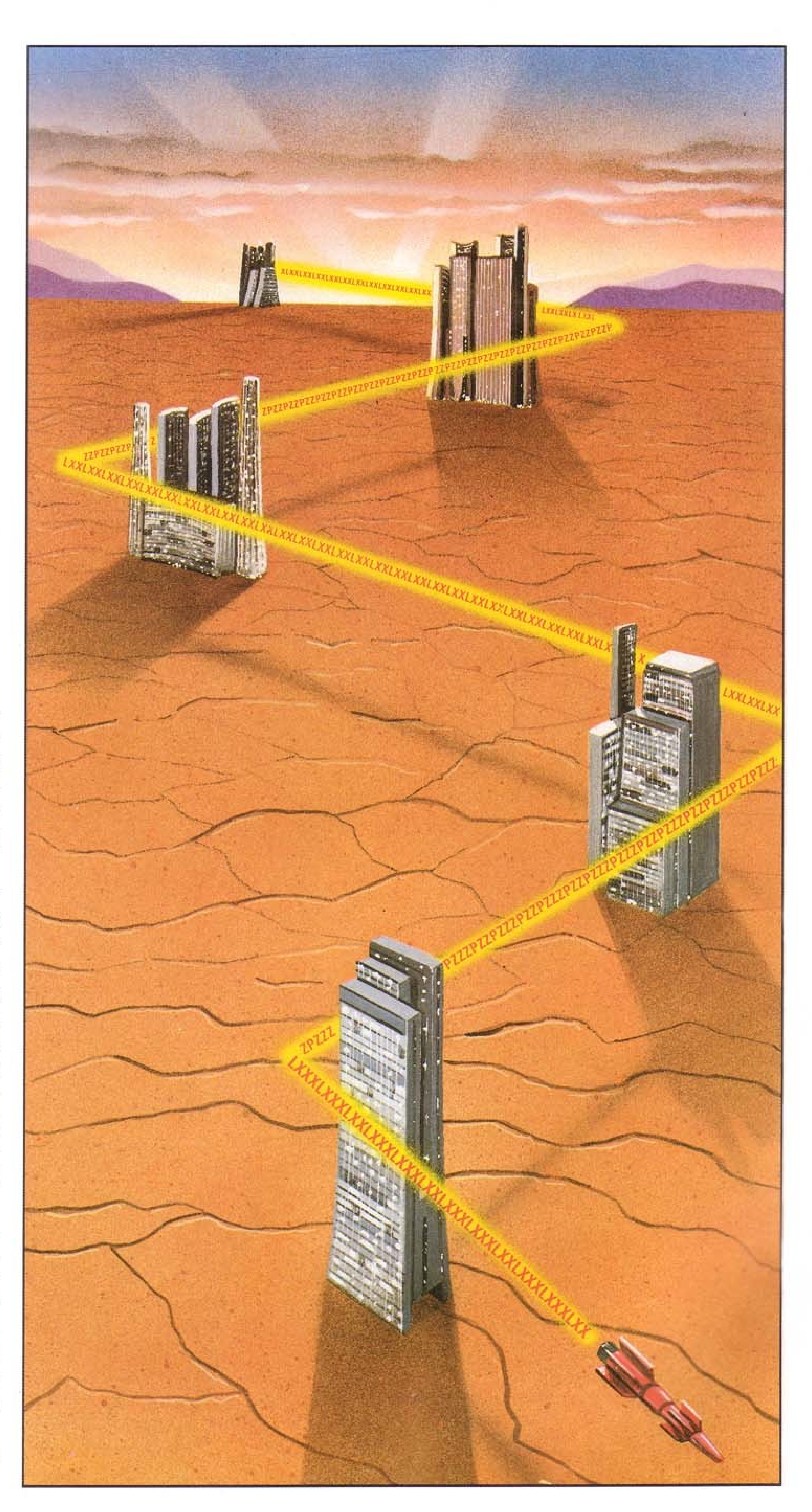
Al ejecutarlo (RUN) verás una estrella cerca de la parte superior de la pantalla. Las teclas de la Z y la X desplazan la base del misil a la izquierda y la derecha, y la tecla F dispara el misil que destruye a la estrella.

Las líneas 130/140 a 200 son similares al anterior programa de disparo del misil. Se han cambiado las variables junto a los GOTOs, pero la única adición es la línea 180. Esta simplemente mira si el misil y la estrella están situados en el mismo espacio. Si lo están, el programa vuelve a comenzar.

La sección intermedia, líneas 80 a 120, es una versión acortada del programa de "desplazamiento por la superficie de la pantalla".

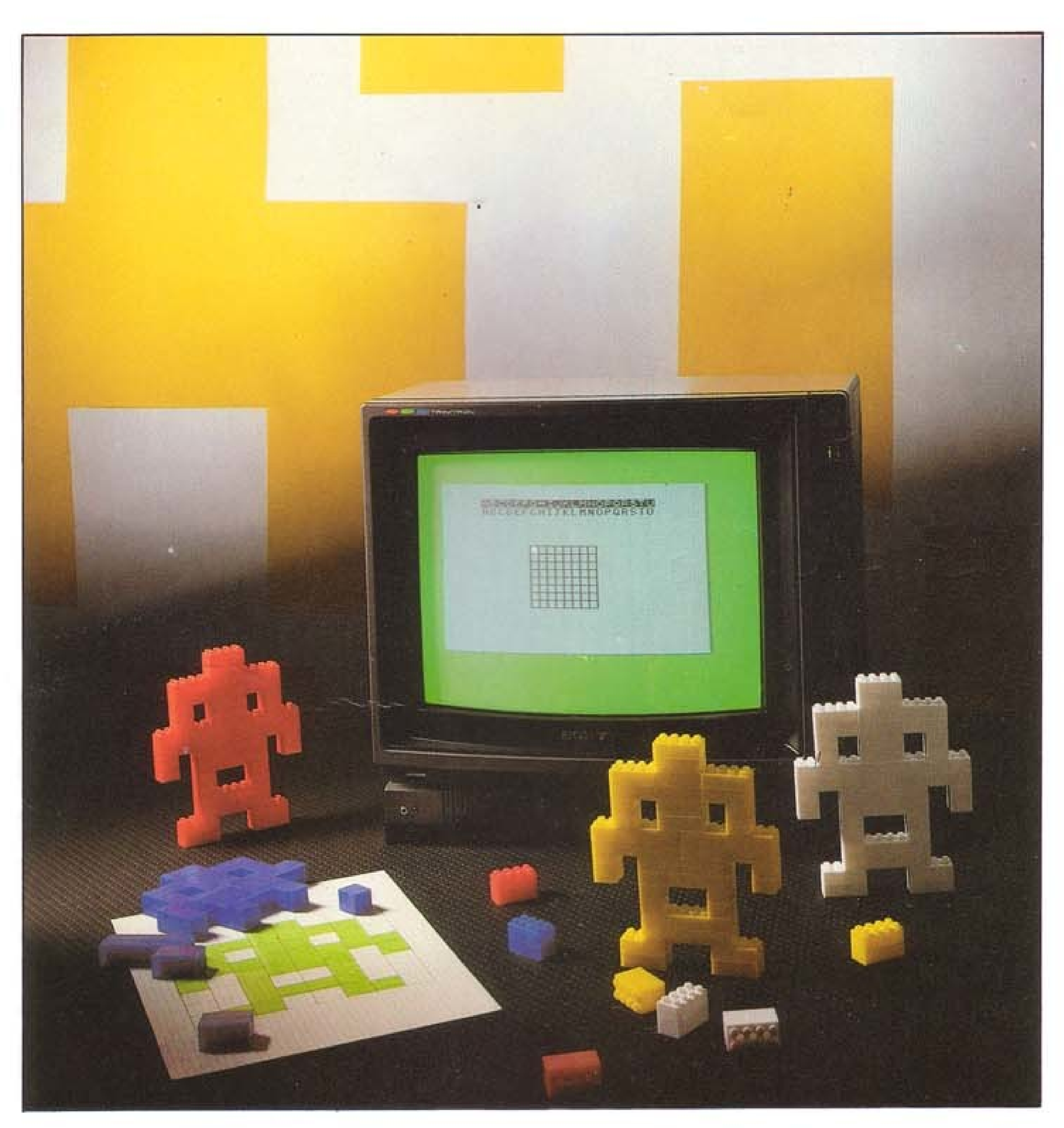
El programa, hasta la línea 70, ejecuta una variedad de funciones. La línea 30 introduce una breve pausa antes de que continúe el programa. Esto es importante cuando la línea 180 completa el bucle al final del programa. Las líneas 40 y 50 ajustan la posición inicial de la base del misil y definen su forma.

Las líneas 60 y 70 eligen el lugar de comienzo y lo visualizan.



DISEÑA TUS CARACTERES GRAFICOS

- ALMACENAMIENTO DE LOS GDU EN UN BANCO DE MEMORIA
- GRABACION DE LOS GDU EN CINTA
- TECLAS DE CONTROL



Si estás cansado del trabajo laborioso que supone la creación de los GDU (Gráficos Definidos por el Usuario) sobre un papel, teclea este programa y tómate un descanso mientras la máquina realiza el trabajo. A continuación podrás dibujar las formas que desees: desde un carácter chino hasta lo último en marcianitos de cuatro patas.

Los GDU son una de las herramientas más versátiles utilizadas por los programadores de gráficos. Pero uno de los inconvenientes que conllevan, si quieres diseñar tus propios dibujos, es el esfuerzo que entrañan: En primer lugar hay que dibujarlos sobre el papel, luego elaborar las sentencias DATA y por último POKEar dentro de la memoria; si además has cometido algún error, te tomará bastante tiempo ver los resultados. Sólo tras la realización de este proceso es posible visionar los GDU en la pantalla de tu monitor.

Por ello presentamos aquí un programa muy distinto a todo lo anteriormente expuesto. Por un lado reemplaza a la hoja de papel cuadriculado por una cuidada presentación en la pantalla, que te permite dibujar cuadraditos (*pixels*) donde desees. En lugar de tener que trabajar a mano dando valores a las sentencias DATA, éstos se obtienen de modo automático.

Pero el valor real del programa radica en que también te permite ver el GDU tal y como está construido —a tamaño natural— evitando la molestia de introducir las DATA dentro de otro programa para esta comprobación. Varias opciones de control permiten reajustar los GDU con la simple presión de una tecla —quizás te gustaría verlo en inverso, o girarlo lateralmente de izquierda a derecha.

Una vez realizado el GDU imaginado desearás que sea posible conservarlo —no olvidemos que dispones de 20 millones de millones de millones de opciones—. De este modo el programa te permite almacenar un banco de GDU terminados al que puedes recurrir cuantas veces precises. También puedes guardar (SAVE) los GDU, sacándolos del programa para tenerlos en una cinta, y cargarlos (LOAD) luego, dentro de cualquier programa, para utilizarlos en la pantalla —u otra vez en el generador de GDU para una edición posterior.

El programa consta de dos partes. La primera parte del listado, que aquí se muestra, nos permite crear el generador básico de GDU. Proporciona una rejilla en el centro de la pantalla, permitiendo movernos por toda ella, para dibujar el patrón de *pixels* que deseamos haya en el GDU final. En la siguiente parte del artículo verás las rutinas que permiten modificar los GDU en aspectos más sutiles, así como ser visualizados en la pantalla.

Te puedes ahorrar tener que teclearlo de nuevo la próxima vez, grabándolo (SAVE) en cinta cuando esté completamente copiado de la revista.

Cuando teclees el programa y lo hagas correr (RUN), aparecerá una rejilla de 8×8 cuadraditos en el centro de la pantalla. Este es el "papel gráfico" sobre el que vamos a planificar nuestro GDU. En él podemos seleccionar qué cuadrados utilizarás para formar el conjunto de *pixels*.

Se dispone de un cursor que toma la forma de un cuadrado en blanco. Este cuadrado no está impreso en un color especial, pero es la versión BRIGHT de cualquier cosa que sobreimprima. De éste destaca, sea cual sea el fondo del cuadrado sobre el que se encuentre. El programa está escrito de forma que el movimiento se realiza con las teclas del cursor, y presionando 0 para activar o desactivar el *pixel*.

Si deas cambiar de teclas para utilizar otras más apropiadas a tus preferencias, simplemente cambia los caracteres incluidos entre comillas de las sentencias IF I$=... en las líneas que van desde la 6520 a la 6555 por aquellas teclas que desees utilizar.

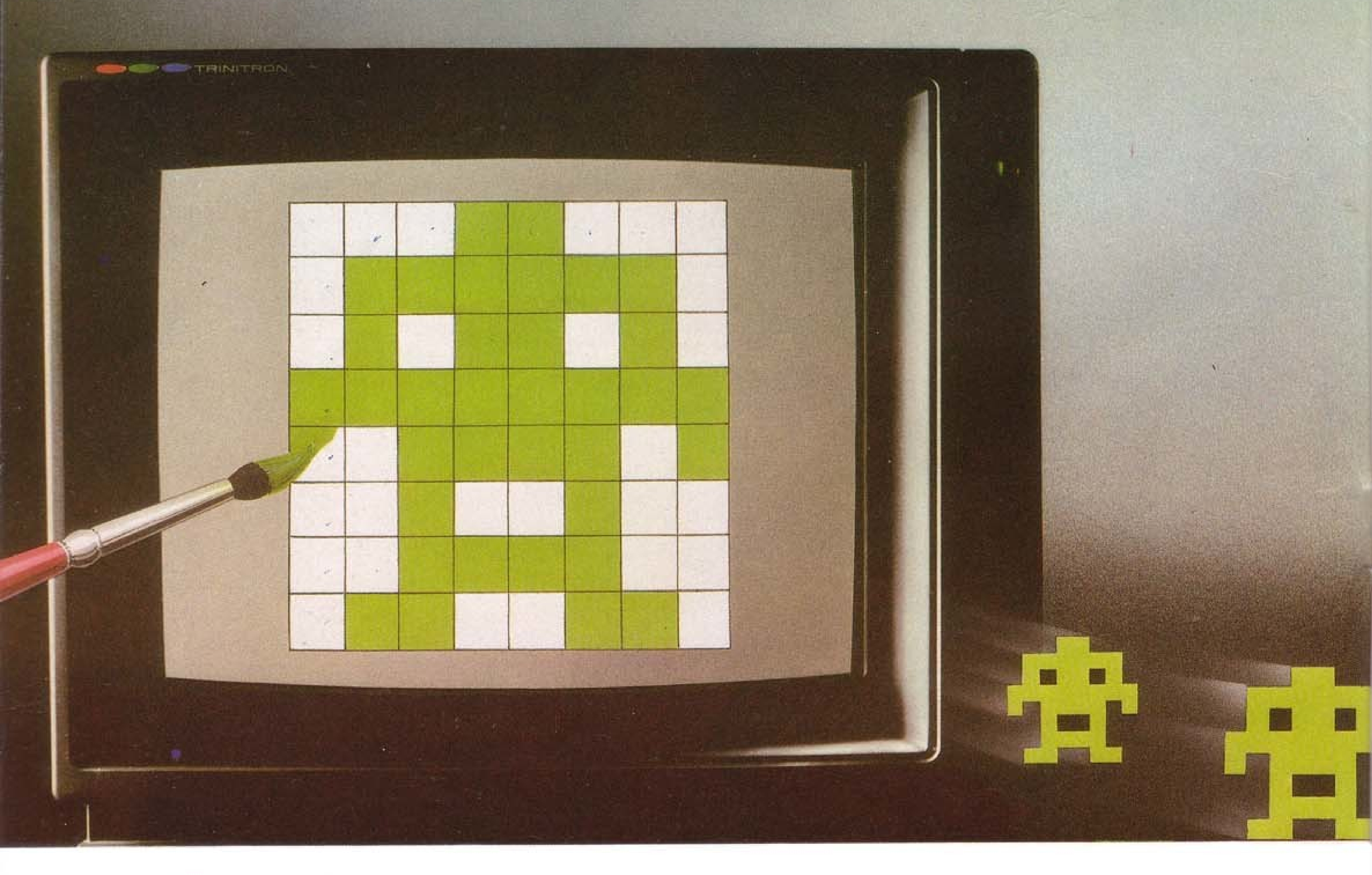
También puede usar este método para hacer que el programa trabaje utilizando algún interface para *joystick*, pero si dispones de un *interface* tipo **Kempston,** necesitarás cambiar algunas líneas. En lugar de la sentencia INKEY$ de la línea 6510, necesitarás utilizar IN. El manual del *interface* resolverá las dudas.

Cualesquiera que sean las teclas que decidas utilizar, debes tener cuidado de no elegir las mismas destinadas a ser teclas de control. Esta parte del programa sólo utiliza tres teclas de control (que son explicadas en el artículo más adelante). Sin embargo la siguiente parte utiliza más. En total las teclas reservadas son C, I, M, P, R, S y T.

Utilizando el teclado o bien un *joystick*, el cursor puede moverse dentro del interior de la rejilla instalada en el centro de la pantalla. Si presionas "fuego", la tecla 0 en el teclado, verás cambiar el color del cursor, que indica que el pixel "bajo" está activado. Si mueves el cursor fuera de esta posición verás que el cuadrado está relleno; de este modo construirás tu diseño.

El programa cuenta con un número de opciones a las que puedes acceder en cualquier momento, mientras editas un GDU. La primera de estas opciones consiste en almacenar el GDU. Para disponer de esta opción, simplemente hay que presionar la tecla S —ésta es una de las teclas de control reservadas que anteriormente mencionamos.

También puede preguntarte qué posición del banco de GDUs deseas que ocupe ese diseño. El banco de GDUs está representado por las letras de la A a la U, mostradas enci-



ma de la rejilla. Cuando hayas introducido una letra, entre la A y la U, para hacer la elección, se producirá un pequeño retraso mientras el ordenador POKEa tu GDU en su lugar. Después lo verás aparecer en el banco, encima de la rejilla. Desde este momento ya puedes continuar.

La siguiente opción es elegir un GDU. Esto se realiza reclamando cualquiera de los GDU del banco para editarlo. Cuando escojas esta opción, presionando la tecla P, el GDU te preguntará qué deseas representar. Tras presionar la tecla apropiada, el GDU correspondiente a esta tecla aparecerá en la rejilla.

GRABACION DE LOS GDU EN CINTA/CASETTE

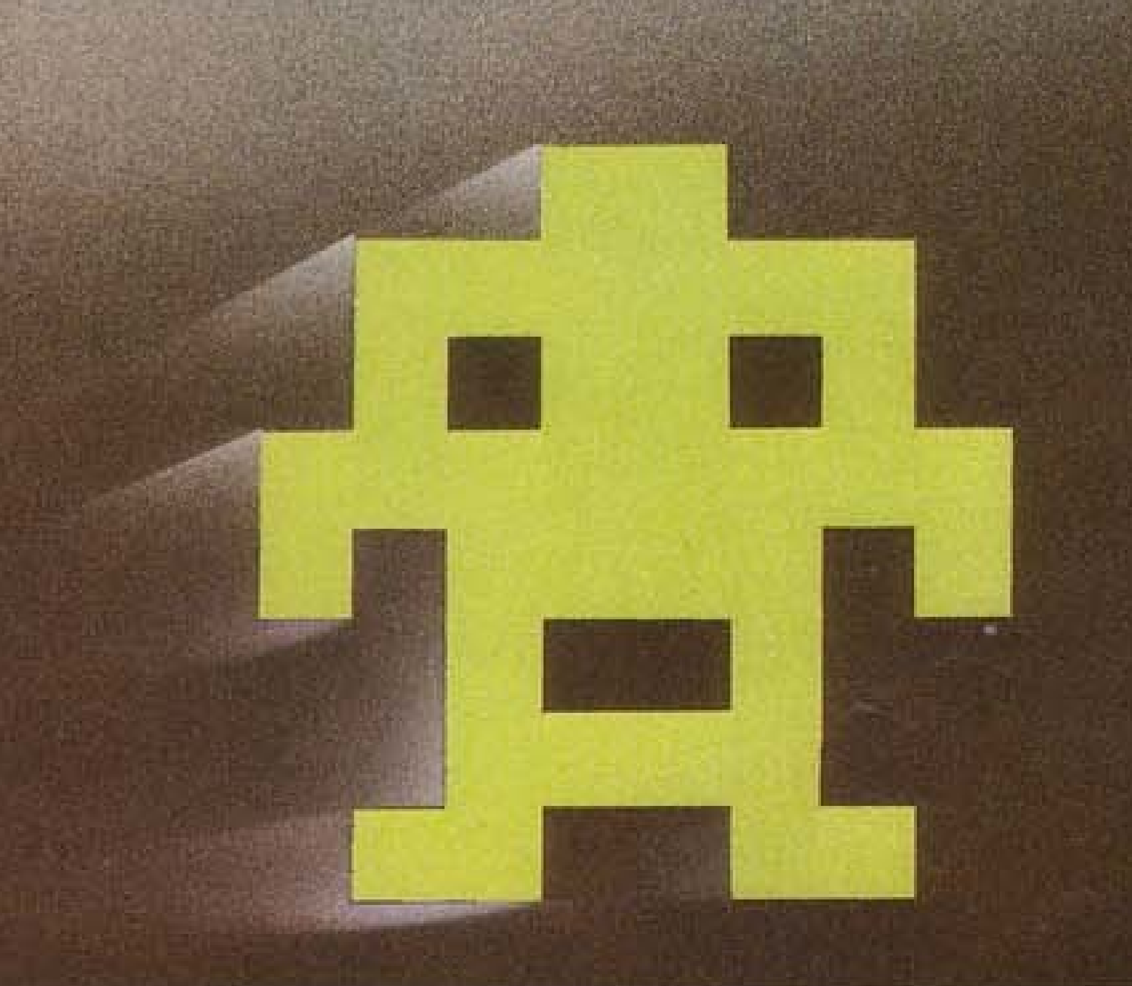
Una vez que el editor del GDU haya realizado su tarea, necesitarás poder grabar (SAVE) los diseños finalizados en la cinta, para utilizarlos en programas propios. Para conseguir esto existe la opción LOAD, de este modo puedes cargar el GDU grabado anteriormente para ser reeditado.

Puedes utilizar fácilmente cualquiera de las opciones, grabar SAVE o cargar LOAD, presionando la tecla T. Se te preguntará qué opción deseas: grabar (SAVE) o cargar (LOAD), y el ordenador lo realizará. Cuando graves (SAVE) en cinta, los *bytes* de los 21 GDUs almacenados son grabados como un bloque de memoria —así que no olvides almacenar el GDU en el que has estado trabajando anteriormente, antes de guardar el banco.

La parte final de este artículo te dará más información acerca de cómo utilizar estas facilidades para diseñar y utilizar muchos GDUs, incluyendo instrucciones detalladas sobre cómo meterlos dentro de otro programa. También añade un número de caracteres extra utilizables.

Teclea

```
   5 CLEAR USR "A"-16
   6 POKE 23675,PEEK
     23675-16
   8 BORDER 4: PAPER 7: INK
     0: CLS
  10 FOR N=USR "A" TO USR
     "B"+7:READ A: POKE N,A:
     NEXT N
  15 POKE 23675,PEEK
     23675+16
  20 POKE 23658,8
  30 LET X=11: LET Y=8: LET
     NX=X: LET NY=Y
 100 LET A$="": FOR N=144 TO
     164: LET A$=A$+CHR$ N:
     NEXT N
 110 LET B$="": FOR N=65 TO
     85: LET B$=B$+CHR$ N:
     NEXT N
1000 FOR N=87 TO 151 STEP 8:
     PLOT N,48: DRAW 0,64:
     NEXT N
1010 FOR N=48 TO 112 STEP 8:
     PLOT 87,N: DRAW 64,0:
     NEXT N
2000 PRINT AT 3,5;A$
2010 PRINT INVERSE 1;AT
     2,5;B$
2400 GO SUB 6500
2500 IF INKEY$="P" THEN GO
     TO 5000
2510 IF INKEY$="S" THEN GO
     TO 5100
2520 IF INKEY$="T" THEN GO
     TO 5200
3900 GO TO 2000
5000 INPUT "ELIGE UN
     CARACTER (A-U)?", LINE
     C$
5010 IF C$<CHR$ 65 OR
     C$>CHR$ 85 THEN GO TO
     5000
5020 LET D=CODE C$+79: LET
     C$=CHR$ D: GO SUB 6000:
     GO TO 2000
5100 INPUT "EN QUE CARACTER
     SE GUARDA (A-U)?", LINE
     C$
5110 IF C$<CHR$ 65 OR
     C$>CHR$ 85 THEN GO TO
     5100
5120 PRINT AT 18,10;
     "ALMACENANDO"
5130 FOR N=USR C$ TO USR C$+7
5140 LET R=0: LET BIT=128:
     FOR M=0 TO 7
5150 IF PEEK (18432+(N-USR
     C$)*32+M+11)<>1 THEN
     LET R=R+BIT
5160 LET BIT=BIT/2: NEXT M:
     POKE N,R
5170 NEXT N: PRINT AT
```



```
     18,Ø;"□";TAB 31;"□":
     GO TO 1ØØØ
5200 INPUT "L(CARGAR) O
     S(GRABAR)?", LINE C$:
     IF C$<>"L" AND C$<>"S"
     THEN GO TO 1ØØØ
522Ø IF C$="L" THEN GO TO
     525Ø
523Ø INPUT "NOMBRE DEL
     FICHERO", LINE N$: IF
     N$="" OR LEN N$>1Ø THEN
     GO TO 523Ø
524Ø SAVE N$CODE USR
     "A",168: GO TO 1ØØ
```



Aspecto que presenta la pantalla.

```
525Ø INPUT "NOMBRE DEL
     FICHERO", LINE N$: IF
     LEN N$>1Ø THEN GO TO
     525Ø
526Ø PRINT AT 19,Ø;: LOAD
     N$CODE USR "A": PRINT
     AT 2Ø,Ø;"□";TAB
     31;"□": GO TO 1ØØ
6ØØØ LET B=USR "A"+8*(CODE
     C$-144)
6Ø1Ø POKE 23675,PEEK
     23675-16
6Ø2Ø FOR N=Ø TO 7
6Ø3Ø LET V=PEEK (B+N)
6Ø4Ø LET BIT=128: FOR M=Ø TO
     7
6Ø5Ø IF V>=BIT THEN PRINT AT
     8+N,11+M;CHR$ 144: LET
     V=V-BIT: GO TO 6Ø6Ø
6Ø55 PRINT AT 8+N,11+M;CHR$
     145
6Ø6Ø LET BIT=BIT/2: NEXT M
6Ø7Ø NEXT N: POKE 23675,PEEK
     23675+16: RETURN
65ØØ POKE 22528+32*Y+X,12Ø:
     PAUSE Ø
651Ø LET I$=INKEY$
652Ø IF I$="5" AND X>11 THEN
     LET NX=X-1
653Ø IF I$="8" AND X<18 THEN
     LET NX=X+1
654Ø IF I$="6" AND Y<15 THEN
     LET NY=Y+1
655Ø IF I$="7" AND Y>8 THEN
     LET NY=Y-1
6552 IF I$="Ø" THEN GO SUB
     7ØØØ
6555 IF I$="" THEN GO TO 658Ø
656Ø POKE 22528+Y*32+X,56
657Ø LET X=NX: LET Y=NY
658Ø POKE 22528+Y*32+X,12Ø
659Ø RETURN
7ØØØ POKE 23675,PEEK 23675-
     16
7Ø1Ø IF PEEK
     (18432+(Y-8)*32+X)=1
     THEN PRINT BRIGHT 1;AT
     Y,X;CHR$ 144: GO TO
     7Ø3Ø
7Ø2Ø PRINT BRIGHT 1;AT
     Y,X;CHR$ 145
7Ø3Ø POKE 23675,PEEK
     23675+16: RETURN
9ØØØ DATA 85,171,85,171,85,
     171,85,255,1,1,1,1,1,1,
     1,255
```



POR FIN EL PLOTTER ES YA ASEQUIBLE... a cualquier ordenador y economía

(Desde 155.166)

PLOTTER ROLAND DXY DE ALTAS PRESTACIONES

- *Formato DIN A3*
- *Pueden operar con inclinación de hasta 60°.*
- *Incluídos interfaces RS-232 y Paralelo Centronics.*
- *Directamente compatible con el software desarrollado para el H.P. 7470 y H.P. 7475.*
- *21 comandos DXY y 50 RD-GL en ROM para usar en sus propios programas.*
- *Set internacional de caracteres incluyendo el Castellano.*
- *Mono o multipluma de 8 colores y gran variedad de tipos y adaptadores.*

DHY SERIES	101	800	880	980
VELOCIDAD	180 mm/sg	180 mm/sg	200 mm/sg	230 mm/sg
RESOLUCION	0,1 mm/paso	0,1 mm/paso	0,05 mm/paso	0,5 mm/paso
PRECISION	± 1%	± 1%	± 0,5%	± 0,5%
REPETICION	≤ ± 0,3 mm	≤ ± 0,3 mm	≤ ± 0,3 mm	≤ ± O,3 mm
P.V.P. pts.	155,166	188,340	247,607	355,918

Solicite información a su distribuidor habitual o directamente en nuestras oficinas.



Magallanes, 14 - Tel.: 446 47 22
Telex 42802 N.B.S.P. - 28015 Madrid

LOGO, LENGUAJE INTUITIVO

Cada vez son más los niños que se introducen en la informática a través del LOGO, un lenguaje diseñado pensando en el aprendizaje. Seymour Papert, **a quien se atribuye un claro conocimiento de la mentalidad infantil, ha introducido una nueva revolución en las aulas con su desarrollo de un lenguaje de alto nivel.**

En 1960 los ordenadores eran muy caros. La potencia de cálculo de tu micro habría costado millones de pesetas, ya que incluso los ordenadores más grandes no podían almacenar más de unos 144 Kbytes. Por razones de economía, los lenguajes de ordenador estaban diseñados para utilizar la menor cantidad de memoria posible, y su concepción pretendía hacerlos fáciles para el ordenador aunque así resultaran más difíciles para el programador.

Con la aparición del microordenador en los años setenta, los lenguajes de programación ganaron en popularidad, ya que los nuevos micros, al igual que los grandes ordenadores de los años sesenta, tenían memorias pequeñas. Cobró cuerpo la idea de que «sencillo para el ordenador» no significaba necesariamente «sencillo para el programador», y se fueron aceptando las dificultades de aprendizaje de lenguajes como el BASIC, como una característica de la programación.

LA ELECCION DE UN LENGUAJE

Al conectar la mayoría de los micros domésticos, éstos trabajan en BASIC (Beginners' All-Purpose Symbolic Instruction Code=Código de Instrucciones Simbólicas de Uso General para Principiantes), siendo éste el lenguaje que domina la mayoría de los propietarios de micros. Sin embargo no hay razón para que esto sea así necesariamente. El BASIC es sólo un programa en código máquina que está presente automáticamente en el ordenador, y de hecho aún existen máquinas para uso doméstico en las que el BASIC ha de cargarse desde una cinta o disco antes de empezar a programar.

Esto significa que es perfectamente posible cambiar el lenguaje que tu ordenador es capaz de entender; todo lo que tienes que hacer es cargar un programa un código máquina que le permita reconocer las instrucciones y realizar las acciones adecuadas. Tal vez ya has descubierto que es relativamente fácil ampliar el BASIC de tu máquina con comandos adicionales.

Pero también es posible no adaptarse meramente al programa existente utilizado por el intérprete BASIC, sino sustituirlo completamente.

Desde los primeros tiempos de la aparición de los ordenadores, se han desarrollado alrededor de unos cien lenguajes diferentes, para diversos fines, además de los múltiples lenguajes "de fabricación casera" diseñados para ordenadores particulares. Algunos de estos lenguajes son tan especializados en sus aplicaciones que nunca te encontrarás con ellos, excepto en los más altos niveles de investigación. Pero hay otros que por lo menos son tan prácticos para el usuario doméstico como el BASIC, y en algunos casos incluso más.

Que puedas disponer o no de un lenguaje particular depende de que puedas obtener o no el programa que permite a tu micro operar con él. Los diferentes lenguajes se suministran normalmente como cualquier otro programa, sobre cinta o disco, y en algunos casos sobre un *chip* de ROM. Su disponibilidad depende del ordenador que tengas. Por ejemplo, muchas máquinas de aplicación comercial que disponen del sistema operativo CP/M pueden elegir entre más de una docena de lenguajes, y en algunos casos entre varias versiones de cada uno. Y existe también una elección relativamente variada para la mayoría de los ordenadores domésticos.

NIVELES DE COMUNICACION

Un lenguaje es una forma de comunicar entre el ordenador y tú. Es



algo que entendéis los dos, un compromiso entre un lenguaje natural (por ejemplo el inglés y el código máquina en binario, que es con lo que realmente trabaja la máquina). Se dice que un lenguaje es de bajo nivel cuando es cercano al lenguaje propio del ordenador. Un ejemplo de ello es el lenguaje ensamblador. Los lenguajes de alto nivel, como el LOGO, pueden estar muy cerca del lenguaje natural. La próxima generación de ordenadores (llamada quinta generación) probablemente utilizará uno de dichos lenguajes para aceptar instrucciones directamente del inglés.

El BASIC se sitúa en algún punto intermedio entre los dos extremos, y según la opinión de muchos programadores no se puede decir que sea un buen compromiso, ya que ni es fácil de entender, ni es rápido para ser usado por el ordenador.

LOS COMIENZOS DEL LOGO

En 1967, un grupo de investigación del **Instituto Tecnológico de Massachusetts** (MIT), estableció unos planteamientos nuevos de acercamiento a los ordenadores. Se propusieron la creación de un lenguaje que resultara fácil para el programador aunque no fuera tan fácil para el ordenador. El resultado fue el LOGO.

El equipo estaba encabezado por **Seymour Papert,** un sudafricano expatriado. **Papert** había trabajado estrechamente con **Jean Piaget,** el famoso psicólogo infantil, según el cual los niños sólo pueden entender un concepto abstracto si se les presenta de una forma concreta. Pensaba que el aprendizaje de un niño debería tener lugar a través de sus propios descubrimientos, en lugar de adoptar una actitud pasiva ante las cosas. Estas ideas tuvieron una gran influencia en el desarrollo del LOGO.



También ejerció influencia el trabajo de **Mary Minsky,** investigadora sobre inteligencia artificial en el mismo instituto durante los años 60. La inteligencia artificial es la ciencia que se ocupa de simular con máquinas los aspectos de la inteligencia humana. Los ordenadores no son inteligentes, sino que sólo obedecen instrucciones que se les han de dar de una forma muy detallada. Los factores que hay que considerar cuando se resuelve un problema son muchos y muy variados, de modo que la simulación de un proceso de este tipo en un programa de ordenador es una tarea gigantesca. El mundo de la inteligencia artificial necesita lenguajes de programación que permitan simular el aprendizaje humano y su capacidad de tomar decisiones. El LISP es un potente lenguaje de programación desarrollado con tal objeto, y del que se hablará más adelante en INPUT. El LOGO es esencialmente un dialecto del LISP, y aunque puede manejar palabras y números, está orientado principalmente a la programación de gráficos.

El nombre de LISP deriva de «List Processing». Su estructura de datos básica no es una matriz de números ni una cadena de caracteres, sino una lista, y dado que una lista puede estar formada por símbolos o por otras listas, es fácil procesar datos no numéricos. Sin embargo el LISP no es fácil de aprender.

DEMOSTRACIONES CONCRETAS

Seymour Papert y sus colegas iban buscando una "puerta de acceso" por la que los niños pudieran entrar al mundo de la programación. Se identificaron así tres áreas de interés para los niños: gráficos, música y robótica. Los niños están interesados en realizar dibujos sobre el monitor

de su ordenador, en servirse de él para crear sonidos electrónicos, y en controlar máquinas desde el teclado. De las tres áreas, la robótica era la más excitante y sugestiva, lo que llevó a **Papert** a crear la tortuga robótica.

El desarrollo de la tortuga se vio estimulado porque en aquella época no había monitores baratos. La tortuga, un robot que se arrastra por el suelo llevando un lápiz, está controlada por el ordenador. El lápiz puede subirse o bajarse a medida que la tortuga se va moviendo, con lo que ésta puede realizar dibujos. Permite que los niños relacionen la geometría con sus propios movimientos al andar o pintar. Con frecuencia se veía a los niños que utilizaban la tortuga moviéndose y probando subrutinas antes de ordenar que lo hiciera la tortuga.

La tortuga se llamó así en honor de **Grey Walter,** un neurólogo y cibernético británico que construyó "tortugas cibernéticas" en los años 50. Se trataba de vehículos accionados eléctricamente que medían el nivel de carga de sus baterías, y cuando estaban bajas se dirigían hacia un dispositivo de carga al que se enchufaban ellas solas. La "tortuga" de **Grey Walter** fue uno de los primeros robots verdaderos.

La tortuga robótica original del MIT casi se extinguió con la aparición de los ordenadores personales y la facilidad de representar cosas en una pantalla de una forma sencilla y barata. Fue sucedida por una versión bidimensional: la tortuga de la pantalla. Se trata de un cursor, representado a veces como un *chevron* y otras veces como una pequeña tortuga. Obedece a las mismas instrucciones que la tortuga robótica, no se rompe y es mucho más barata.

Sin embargo la tortuga robótica ha hecho su reaparición en las escuelas, ya que permite que participen más niños en las actividades de programación, proporcionando un modelo para la «geometría de cuerpos», y una excitante introducción concreta a un mundo abstracto.

Estos robots se están haciendo muy populares en las escuelas elementales y están cubriendo su objetivo original de introducir a los niños en la programación de ordenadores de una forma divertida y comprensible.

HABLANDO EL LENGUAJE

¿Qué tienen que ver los juguetes mecánicos con un lenguaje de ordenador? **Papert** considera el ordenador como un vehículo para la creatividad y la expresión de ideas. Piensa que la mejor manera de aprender cosas acerca de los ordenadores es crecer en una cultura del ordenador, de la misma forma que la mejor manera de aprender italiano es pasarse una temporada en Italia. En la conferencia anual correspondiente a 1983 de la principal asociación profesional de ordenadores de América, pidió que se buscase una fórmula para darle un ordenador a cada niño americano. Piensa que es el niño quien debe programar al ordenador y no el ordenador quien programe al niño. Los medios que propone para lograr esto pasan por el LOGO, que según su punto de vista da a los niños control sobre uno de sus más potentes recursos, proporciona una base de partida para resolver problemas fuera del cálculo, y permite que se presenten de una forma sencilla e inteligible las ideas matemáticas complejas.

Papert explicó su filosofía en su célebre libro "Frenesí mental, Niños, Ordenadores e Ideas Potentes". Desde la publicación de su libro en 1980, han aparecido versiones del LOGO para la mayoría de los ordenadores domésticos. Está disponible para diversos microordenadores y la mayor parte de las versiones se parecen estrechamente al LOGO original del MIT. También hay en el mercado varios programas con nombres tipo LOGO, como el «Logo Dart» y el «Logo Graphics».

Se trata de simulaciones de la tortuga gráfica, que sólo constituye una pequeña parte del LOGO. El lenguaje tiene utilidades completas para el proceso de listas y palabras, funciones matemáticas y de sonido y muchas otras prestaciones que no aparecen en los programas "Pseudo-Logo".

El LOGO es el primer lenguaje "próximo al usuario". Como es sencillo de aprender y abunda en los colegios, se piensa con frecuencia que es "para chicos". Nada más lejos de la verdad. El profesor **Harold Abelson,** uno de los diseñadores del LOGO en el MIT, declara: "Al trabajar con el LOGO hemos descubierto algunas cosas importantes."

"Un lenguaje de ordenador puede ser sencillo y potente al mismo tiempo. De hecho no se trata de aspectos conflictivos sino complementarios, ya que es la falta de potencia expresiva en los lenguajes primitivos tales como el BASIC, lo que hace tan difícil para los principiantes el escribir programas simples que hagan algo interesante. Lo que es más importante, hemos visto que es posible dar a la gente control sobre potentes recursos informáticos, que pueden usar como herramientas aprendiendo, jugando y explorando."

El LOGO es también un lenguaje que crece. La universidad de Edimburgo está desarrollando una versión llamada Control-LOGO, que permita un control más sofisticado de los robots. **Papert** pretende que las futuras versiones incluyan "mundos" en los que los niños puedan jugar con las ideas de la Física, de la misma forma que juegan con la Geometría con la Tortuga Gráfica.

PROGRAMANDO EN LOGO

Al cargar el LOGO desde un *cassette* o un *chip* de ROM, te enviará un mensaje de este tipo:

```
BIENVENIDO AL LOGO

?
```

La interrogación (?) es un anunciador o invitación a que introduzcas algo. El LOGO está esperando a que se le dé alguna orden. Si tecleas:

```
HOLA LOGO, TE HE ESTADO
  BUSCANDO
```

¡PARTICIPA EN INPUT!

Si quieres ver tus programas, ideas, o artículos, publicados en tu revista, examina las bases y haznos llegar el material.

Publicar tiene su recompensa.

BASES

PROGRAMAS: Una vez desarrollado tu programa, que debe ser original y no haber sido enviado a ninguna otra publicación, puedes enviárnoslo aquí grabado en cassette, diskette o microdrive. Es preferible que vaya acompañado por un listado de impresora, pero no es imprescindible.

El programa habrá de venir acompañado por un texto que aclare cuál es su objetivo, el modo de funcionamiento y una explicación del cometido que cumplen las distintas rutinas que lo componen. El texto se presentará en papel de tamaño folio y mecanografiado a dos espacios. No importa que la redacción no sea muy clara y cuidada; nuestro equipo de expertos se encargará de proporcionarle la forma más atractiva posible.

ARTICULOS E IDEAS: Se aplica lo anteriormente dicho para los textos que acompañan a los programas; es decir, conviene detallar al máximo lo que desees que aparezca publicado en la revista, de la manera que te gustaría que otra persona hubiera explicado eso mismo.

UN JURADO propio decidirá en cada momento qué colaboraciones reúnen los requisitos adecuados para su publicación, y evaluará la cuantía del premio en metálico al que se hagan acreedoras.

No olvidéis indicar claramente para qué ordenador está preparado el material, así como vuestro nombre y dirección y, cuando sea posible, un teléfono de contacto. Entre todos los trabajos recibidos durante cada mes SORTEAREMOS:

- **Un premio de 50.000 ptas.**
- **Un premio de 25.000 ptas.**
- **Un premio de 10.000 ptas.**
  en material microinformático a elegir por los afortunados.

¡No os desaniméis!, por muy simples o complejas que puedan parecer vuestras ideas, todas serán revisadas con el máximo interés.

INPUT SINCLAIR
Alberto Alcocer, 46, 4.° B
28016 Madrid

NOTA: INPUT no se responsabiliza de la devolución del material que no vaya acompañado por un sobre adecuado con el franqueo correspondiente.

El LOGO responderá con:

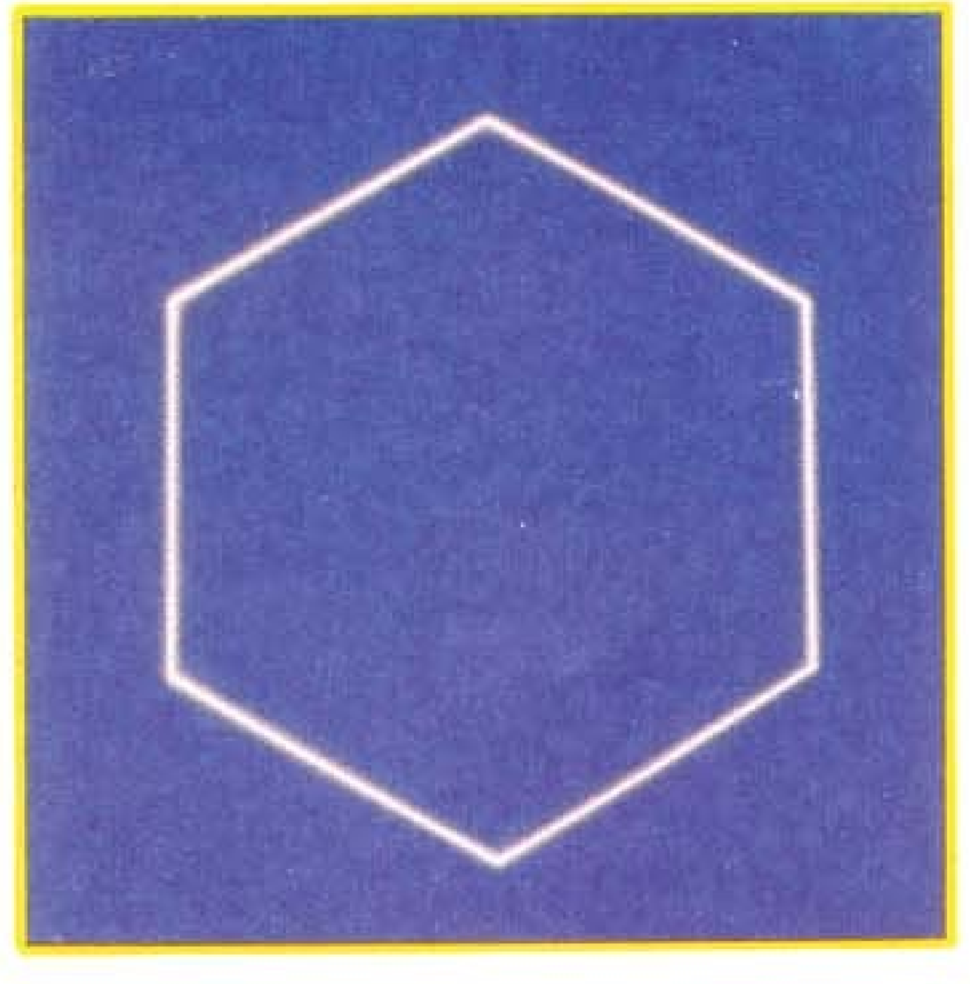
```
NO SE COMO HACER HOLA

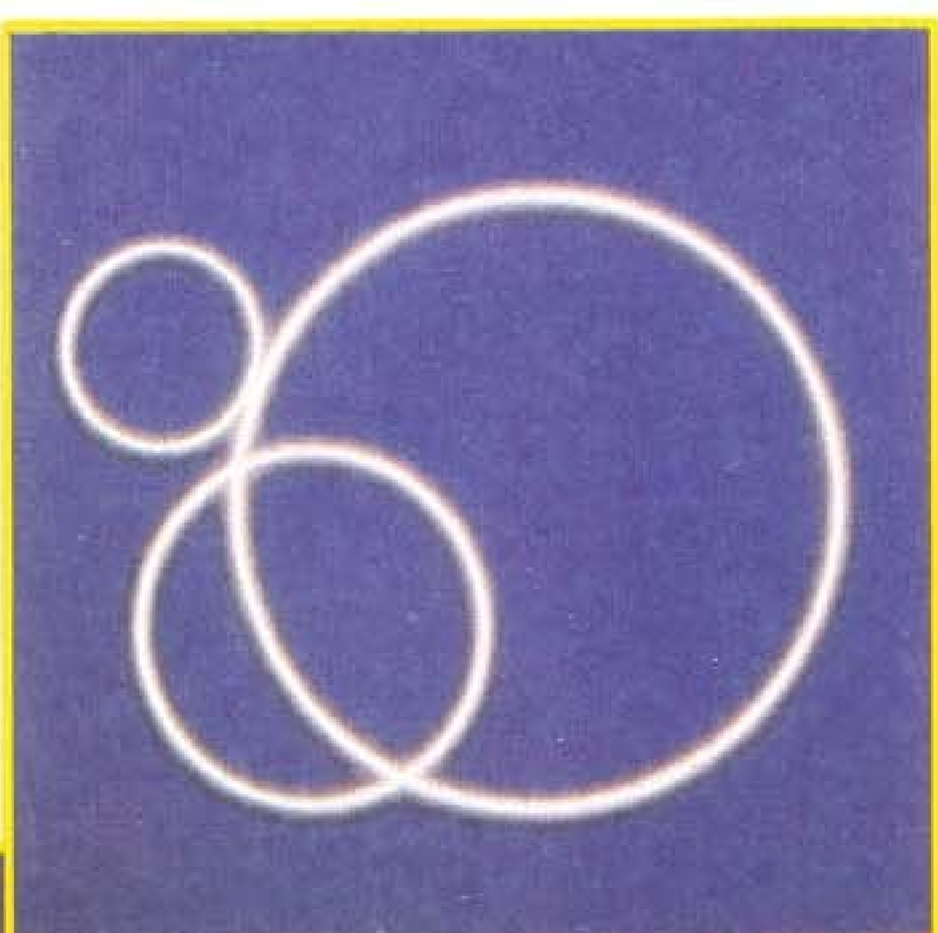
?
```

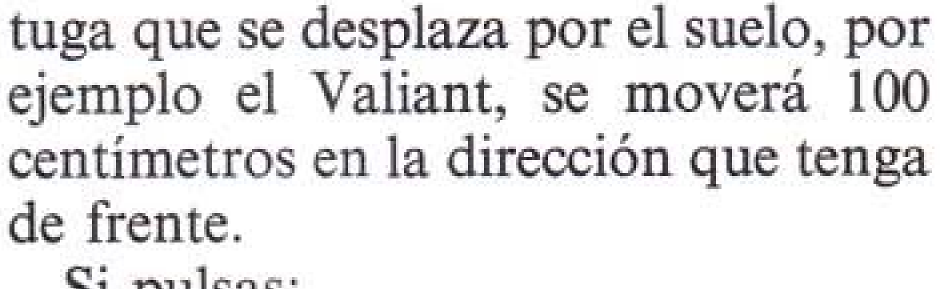
Ha mirado la primera palabra sin identificarla como instrucción, y te ha informado de ello.

Un comando que sí reconocerá es ST, que quiere decir mostrar la tortuga (Show Turtle). Llamará a la tortuga desde las profundidades del ordenador hasta la pantalla. La forma de la tortuga aparecerá en una dirección particular. Este es su encabezamiento, aparecerá provista de un lápiz y listo para pintar.

La instrucción para mover la tortuga hacia adelante es FORWARD, que se puede abreviar a FD. FORWARD es una instrucción del LOGO que requiere una entrada. Hay que decirle a la tortuga cuánto tiene que moverse. Si se pulsa FORWARD 100, la tortuga se moverá 100 unidades hacia adelante. Dejará tras de sí una línea. Si está utilizando una tor-







tuga que se desplaza por el suelo, por ejemplo el Valiant, se moverá 100 centímetros en la dirección que tenga de frente.

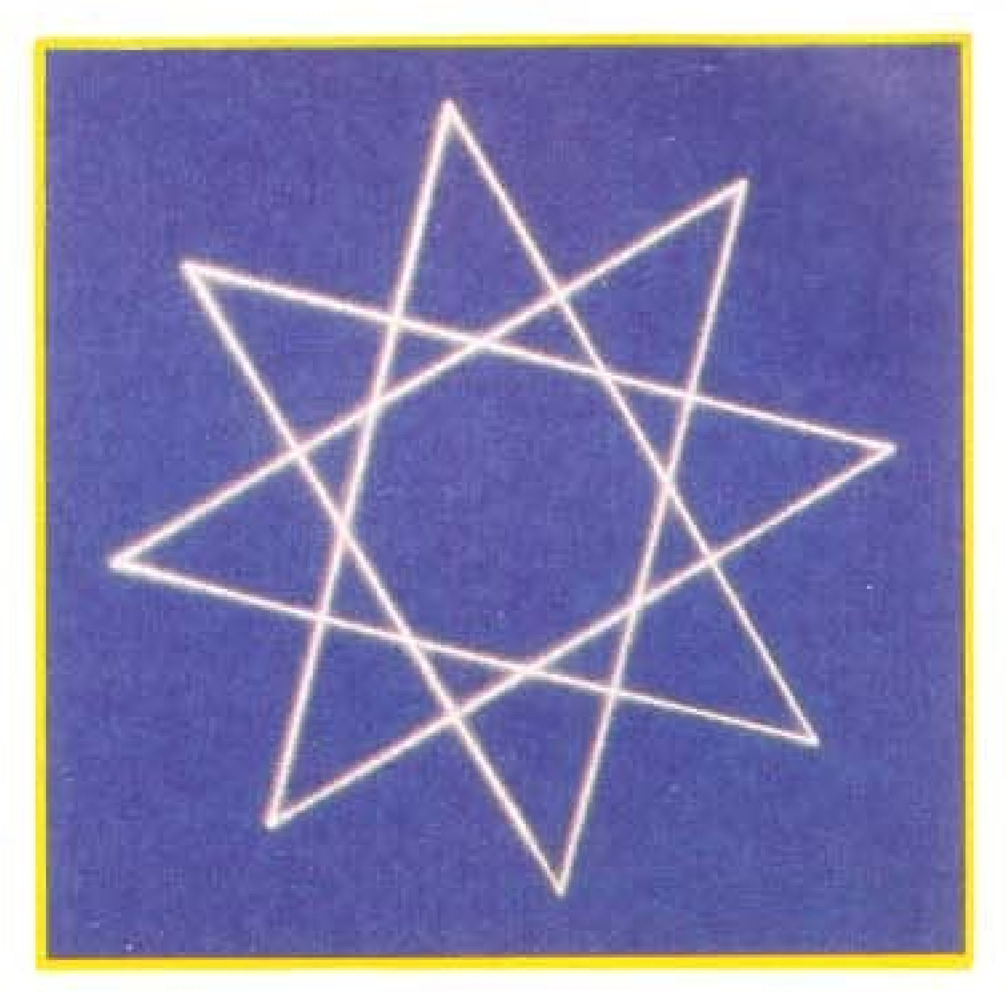
Si pulsas:

```
FORWAR1ØØ
```

El LOGO responderá

```
NO SE COMO HACER FORWAR1ØØ
```

Esto se debe a que, al no haber intercalado un espacio entre el comando y la cantidad numérica, el





LOGO considera FORWARD100 como una palabra diferente de FORWARD, y dado que FORWARD100 no forma parte de su vocabulario envía un mensaje de no reconocimiento. Lo mismo ocurriría si en vez de FORWARD hubieras introducido FERWARD.

Hay un sencillo editor de línea que permite corregir los errores antes de pulsar [ENTER]. Las teclas del cursor permiten trasladarse a lo largo de la línea hasta llegar a la falta. La tecla de borrado (delete) elimina el carácter situado a la izquierda del cursor. Para insertar nuevos caracteres no tienes más que teclearlos. El texto que hay más a la derecha se moverá automáticamente para dejar sitio al texto insertado. Algunas versiones de LOGO permiten recuperar y editar una línea después de haber pulsado [ENTER].

La instrucción BACK trabaja de la misma forma que FORWARD. Su abreviatura es BK. Hace que la tor-

SEPTIEMBRE DE 1985 LA MEJOR 695,- PTAS.

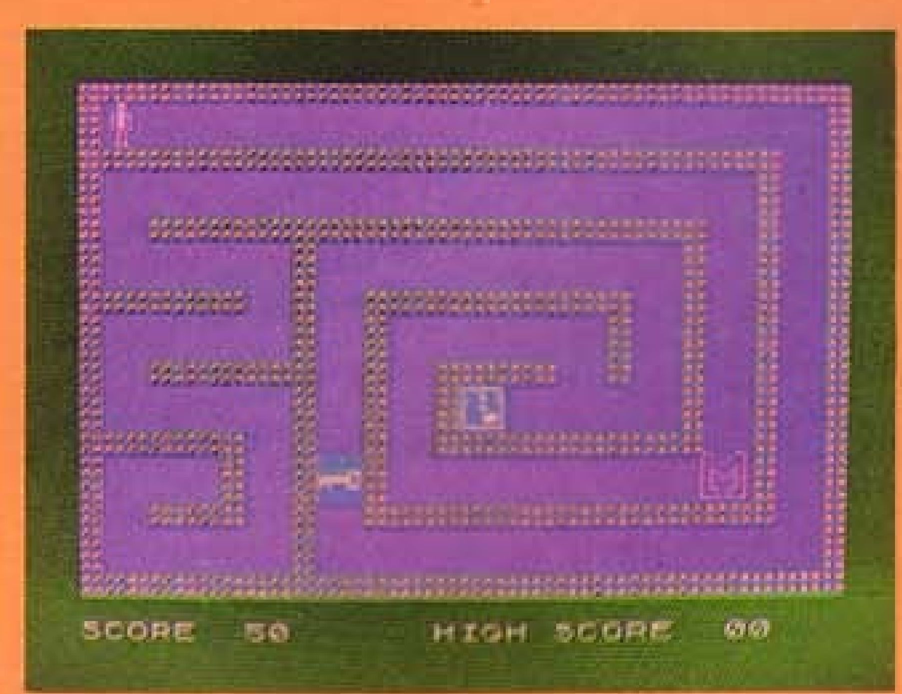
Primera Revista Española en Cassette

SPECTRUMANIA

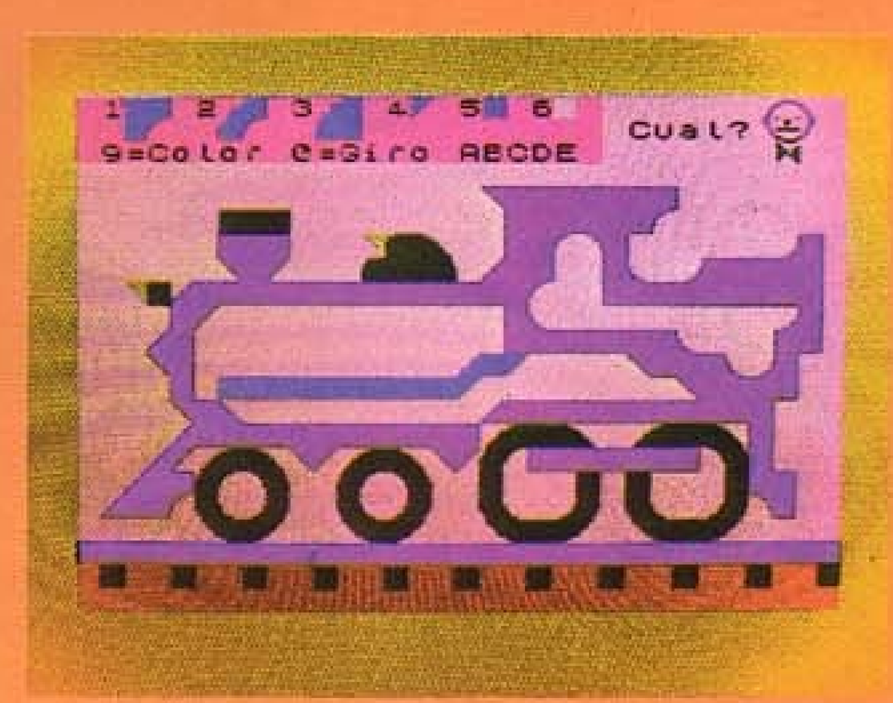
para Spectrum 16K ó 48K

3

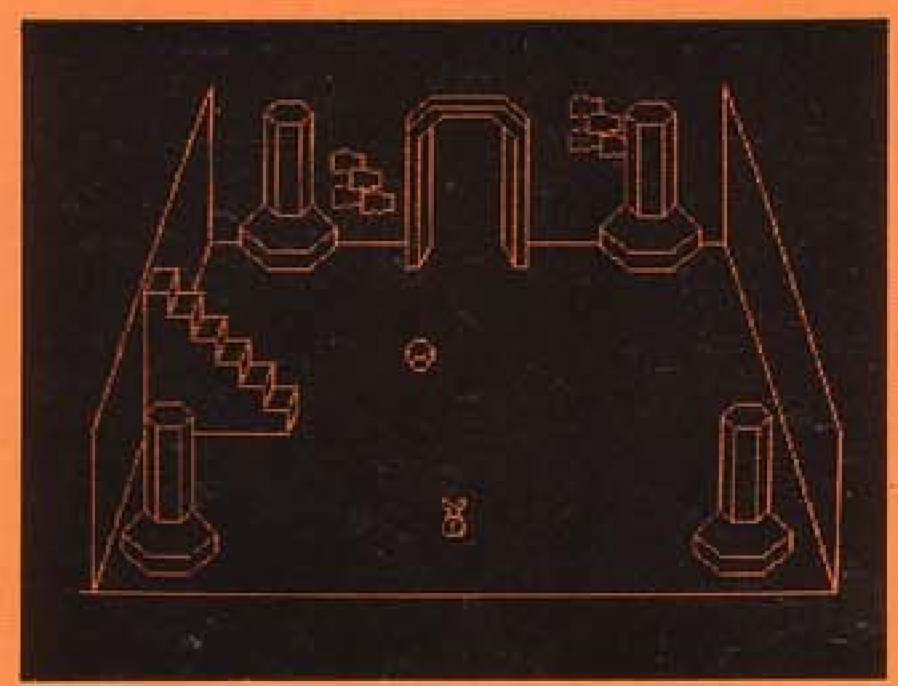
II época



MUTACION: Introdúcete en los circuitos de tu ordenador y lucha contra el «microprocesador asesino».



CONSTRUCTOR: Cualquier diseño, por complicado que parezca, puede hacerse realidad.



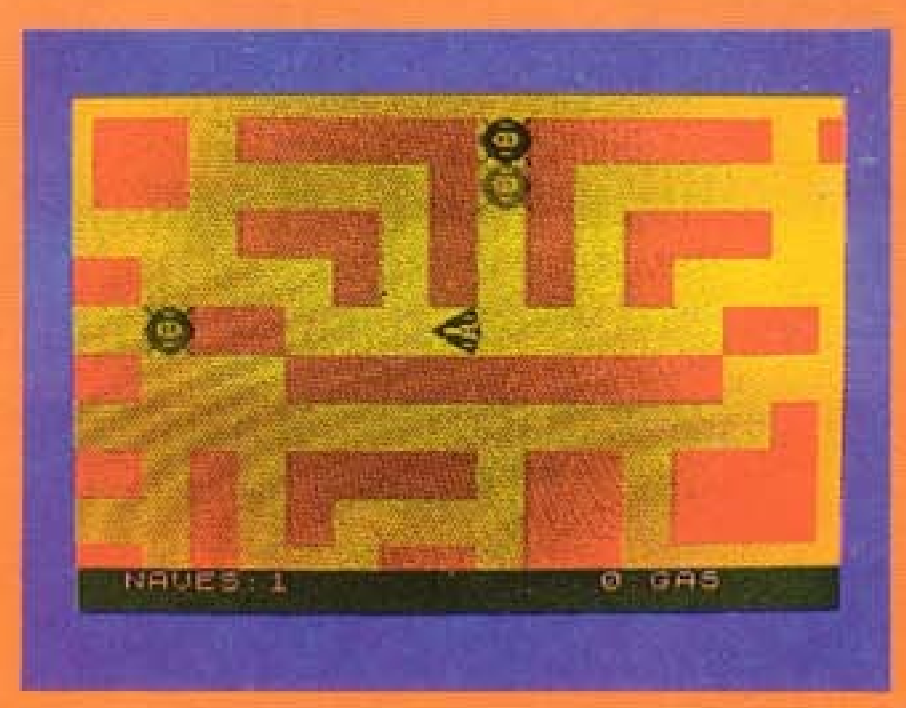
ZEUS: Lucha con tu potente láser contra la plaga que invade el templo de Zeus.



AUTOSTOPISTA GALACTICO: Flotando en la ingravidez del espacio tropezarás con los más curiosos objetos.

ADEMAS

- TCHUQUILAN
- SPEAKER SYSTEM
- CONCURSO DE CARTAS
- CARTERO LOCO
- FRUIT MACHINE
- CONTROL STOCKS
- USER GRAPHICS
- SPRITES
- COMENTARIOS, PROGRAMAS Y PANTALLAS DE:
  - SITI CALC
  - ABU SIMBEL
  - SPECTRUM TOP HITS Y... MUCHO MAS



EL LABERINTO INTERESTELAR es un desafío para cualquier guerrero del espacio.

¡YA ESTA A LA VENTA! COMPRALA EN TU QUIOSCO HABITUAL

BOLETIN DE PEDIDO
Enviar a: VENTAMATIC - C/Corcega, 89, entlo – 08029 BARCELONA – Tel. 230 97 90

Deseo:

☐ Recibir el N.° 3 (2.ª época) / N.° 2 (2.ª época) / N.° 1 (2.ª época) de SPECTRUMANIA, al precio de 695,– ptas. cada uno.

☐ Recibir el N.° 1 (1.ª época) / N.° 2 (1.ª época) de SPECTRUMANIA, al precio de 500,– ptas. cada uno.

☐ Suscribirme por 6 números a la revista SPECTRUMANIA, a partir del N.° ________, al precio de 4.000,– ptas.

☐ Recibir el CATALOGO COMPLETO VENTAMATIC (32 páginas) de artículos de micro-informática, al precio de 200,– ptas., a deducir de mi próximo pedido a VENTAMATIC.

ATENCION: Las personas que se suscriban por 6 números de SPECTRUMANIA antes del 30 de julio de 1985, recibirán un PROGRAMA-SORPRESA DE REGALO.

Fecha: ________
Nombre: ________
Apellidos: ________
Dirección: ________
Población: ________
Provincia: ________ D.P.: ________

Señalar con una cruz la forma de pago:

☐ Talón adjunto (sin gastos de envío).
☐ Giro Postal N.° ________ (sin gastos de envío).
☐ Contra-Reembolso (+ 200,– ptas. de gastos de envío).
☐ Tarjeta VISA / MASTERCARD / AMERICAN EXPRESS (+ 200,– ptas. de gastos de envío), N.° ________
Caduca: ________ FIRMA:

GARANTIA DE CARGA

7 VIDEOJUEGOS Y 3 UTILIDADES EN CODIGO MAQUINA, 1 PROGRAMA DE GESTION, 1 PROGRAMA EDUCATIVO, COMENTARIOS DE SOFTWARE, CONCURSOS.

OFERTA ESPECIAL
REGALO-SORPRESA
Vale por 10 % de DESCUENTO y un
en tu próxima compra
de HARDWARE y SOFTWARE
directamente a VENTAMATIC
(este descuento no es aplicable
a ejemplares de
SPECTRUMANIA)
reproducción sin valor

CONTIENE REGALOS SORPRESA
2 Wafadrives y 200 Programas
GRATIS
¡Busca en tu pantalla!



tuga se mueva hacia atrás, y, como antes, requiere una entrada numérica. Puedes dar cualquier número como entrada para FORWARD o BACK. Así cambiará la posición de la tortuga, pero no su orientación.

Para girar la tortuga, utiliza LEFT y RIGHT. Se pueden abreviar por LT y RT. Al igual que FORWARD y BACK, requieren una entrada numérica. Por ejemplo, RIGHT 39 hará girar la tortuga 39 grados a la derecha. Análogamente LEFT 123, hará que la tortuga gire 123 grados hacia la izquierda.

ALGUNOS JUEGOS SENCILLOS

Si se utiliza el LOGO con niños, es mejor no decirles que RIGHT 96 hace girar la tortuga 96 grados hacia su derecha. Hay que dejar que experimenten con diferentes valores numéricos de entrada y descubran por sí mismos el efecto de los diferentes valores. Se pueden ensayar diversos juegos encaminados a que descubran los valores de diferentes ángulos.

Se puede marcar una posición en la pantalla y ver cuántas instrucciones necesita un niño para detener la tortuga debajo de la marca. O se puede dibujar una carretera en la pantalla y hacer que el niño haga avanzar la tortuga, perdiendo un punto por cada instrucción y un punto cada vez que la tortuga se salga de la carretera. Más adelante daremos en este artículo un procedimiento para dibujar una carretera.

El programa se puede almacenar en disco o cinta, y cargarse cada vez que los niños quieran jugar con él. Si quieres jugar inmediatamente, dibuja la ruta directamente sobre la pantalla del televisor, utilizando un lápiz de cera, o un rotulador de los usados para pintar sobre pizarra blanca. Se puede borrar con facilidad al terminar. De esta forma es más fácil dibujar rutas más complicadas.

Hay una variedad de juegos que se pueden practicar con una tortuga de suelo, basados en recorrer laberintos, tirar cosas, recoger objetos en diversos puntos, etc. Se puede jugar a la tortuga empujona, una variante del juego de los bares, que utiliza una tortuga en vez de una moneda.

Es posible jugar a la tortuga empujona en la pantalla, escribiendo un procedimiento para dibujar la tortuga, y colocándola al principio.

MAS COMANDOS

Para jugar con las instrucciones de traslación y giro se requieren algunas otras instrucciones de LOGO o **primitivas.** Si la pantalla no está en modo "enrollado", es decir, si la tortuga desaparece por la parte superior pero no reaparece por la inferior, necesitas hacer que vuelva a aparecer.

HOME hará que la tortuga regrese a su posición y orientación originales en el centro de la pantalla.

También necesitas poder borrar la pantalla para hacer nuevos dibujos.

La instrucción CLEARSCREEN, que se abrevia CS, borrará todos los dibujos y situará la tortuga en su posición HOME.

A veces desearás que la tortuga se mueva sin dibujar. La instrucción PENUP, abreviada PU, levanta el lápiz de la tortuga. Si utilizas una tortuga de suelo, se levanta el lápiz que lleva en la panza. La tortuga de la pantalla simplemente deja de dibujar.

La instrucción PENDOWN, abreviada PD, hace que baje el lápiz de la tortuga de suelo y que la tortuga de pantalla dibuje de nuevo.

Las instrucciones CLEARSCREEN, HOME, SHOWTURTLE, PENUP y PENDOWN son primitivas de LOGO que no requieren entradas.

He aquí un ejemplo ilustrativo sobre estas primitivas. Al final de cada línea has de pulsar la tecla ENTER.

```
SHOWTURTLE
LEFT 45
FORWARD 71
RIGHT 135
PENUP
FORWARD 5Ø
PENDOWN
LEFT 45
BACK 71
PENUP
HOME
PENDOWN
```

Si quieres intentar alguna otra cosa, puedes borrar el dibujo de la pantalla utilizando la instrucción CLEARSCREEN.

CONSTRUCCION DE UN PROCEDIMIENTO

Hasta aquí todas las actividades descritas han sido del modo 'inmediato". Has estado hablando directamente a la tortuga, que ha ejecutado inmediatamente tus órdenes, igual que un pelotón de soldados obedeciendo a un sargento instructor en una plaza de armas. Hay otro modo de funcionamiento en LOGO, el modo de procedimiento.

En el modo de procedimiento se nombra una serie de comandos, los cuales se escriben después del nombre. De esta forma el nombre asignado al procedimiento pasa a formar parte del vocabulario del LOGO. La tortuga responderá a este nombre ejecutando los comandos que figuran en su definición. Para definir un procedimiento, utiliza TO seguido del nombre elegido. Se puede utilizar cualquier nombre excepto el de una primitiva de LOGO existente. Las cosas resultan más fáciles si asignas a los procedimientos nombres que definan su función.

Se entenderá mejor esto si ponemos un ejemplo. He aquí un procedimiento que enseña a la tortuga a dibujar un zig-zag:

```
TO ZIGZAG
FORWARD 2Ø
LEFT 15Ø
FORWARD 2Ø
RIGHT 15Ø
FORWARD 2Ø
LEFT 15Ø
FORWARD 2Ø
RIGHT 15Ø
FORWARD 2Ø
LEFT 15Ø
FORWARD 2Ø
RIGHT 15Ø
END
```

TU PUEDES AYUDAR A ETIOPIA

RECHAZANDO LAS COPIAS PIRATAS



Después de teclear TO ZIGZAG seguido de ENTER la invitación o *prompt* cambia de ? a >. Esto te indica que estás en modo procedimiento. Al terminar el procedimiento, pulsa END y el anunciador volverá a ser ?, lo que significa que nuevamente estás en modo inmediato.

ZIGZAG forma ahora parte del vocabulario del LOGO. Si lo tecleas, la tortuga dibujará un zig-zag.

El LOGO tiene una instrucción de repetición que se puede utilizar para teclear varias veces la misma cosa. Se podría escribir de nuevo ZIGZAG como:

```
TO ZIGZAG
REPEAT 3 [FORWARD 2Ø LEFT 15Ø
  FORWARD 2Ø RIGHT 15Ø]
END
```

La rutina que ha de repetirse aparece encerrada en paréntesis cuadrados, precedida por REPEAT y el número de veces que ha de ser repetida. Si piensas en un círculo como en una serie de traslaciones y giros cortos, REPEAT facilita el dibujo de curvas. Por ejemplo REPEAT 180 [FORWARD 1 RIGHT 1] dibujará un semicírculo.

PONIENDOLO TODO JUNTO

Ya tienes todos los ingredientes para dibujar una pista de carreras. La manera de resolver un problema con el LOGO es dividirlo en lo que **Seymour Papert** llama "Bytes a medida de la mente". Así, para empezar con el borde interior, dibuja primero una curva:

```
REPEAT 18Ø [FORWARD 1
RIGHT 1]
```

Y ahora una línea recta:

```
FORWARD 1ØØ
```

Puedes combinar dos curvas y dos bordes rectos en un procedimiento para dibujar el interior de la pista:

```
TO INTERIOR
REPEAT 2 [FORWARD 1ØØ REPEAT
 18Ø [FORWARD 1 RIGHT 1]]
END
```

Es perfectamente legítimo tener en REPEAT dentro de otro REPEAT siempre que te acuerdes de cerrar todos los paréntesis al final.

Para el exterior de la pista, necesitas una curva más amplia, creada aumentando el tamaño de los pasos de la tortuga.

```
REPEAT 18Ø [FORWARD 2
  RIGHT 1]
```

es demasiado grande, pero algo cercano a:

```
REPEAT 9Ø [FORWARD 3 RIGHT 2]
```

estará bastante bien.

Vayamos ahora al exterior de la pista.

```
TO EXTERIOR
REPEAT 2 [FORWARD 1ØØ REPEAT
  9Ø [FORWARD 3 RIGHT 2]]
END
```

Pero si tecleas:

```
EXTERIOR
INTERIOR
```

no tendrás el resultado ideal.

Puedes escribir un procedimiento para empezar a dibujar en un sitio más adecuado, mediante un procedimiento llamado BEGIN:

```
TO COMENZAR
PENUP
LEFT 4Ø
FORWARD 11Ø
RIGHT 13Ø
PENDOWN
END
```

Otro procedimiento moverá la tortuga a una posición adecuada para dibujar la parte interior de la pista, proporcionando además una línea de partida.

```
TO MOVER
RIGHT 9Ø
FORWARD 3Ø
LEFT 9Ø
END
```

Para ver la pista debes llamar a todos los procedimientos a la vez, tecleando:

```
COMENZAR
EXTERIOR
MOVER
INTERIOR
```

Ahora puedes escribir un procedimiento que coloque a la tortuga en la línea de salida:

```
TO PRINCIPIO
LEFT 9Ø
FORWARD 15
RIGHT 9Ø
END
```

Una vez que un procedimiento está ya en la memoria de un ordenador, el LOGO te permite usarlo de la misma forma que cualquier primitiva de LOGO. Esto significa que puedes usar los procedimientos para ayudar a definir otros nuevos procedimientos. Así, puedes combinar los bloques constructivos que has llamado BEGIN, OUTSIDE, MOVE, INSIDE y START, en un nuevo procedimiento que denominaremos GAME:

```
TO JUGAR
COMENZAR
EXTERIOR
MOVER
INTERIOR
PRINCIPIO
END
```

Cada vez que tecleas GAME, aparecerá la pista con la tortuga en la línea de salida.

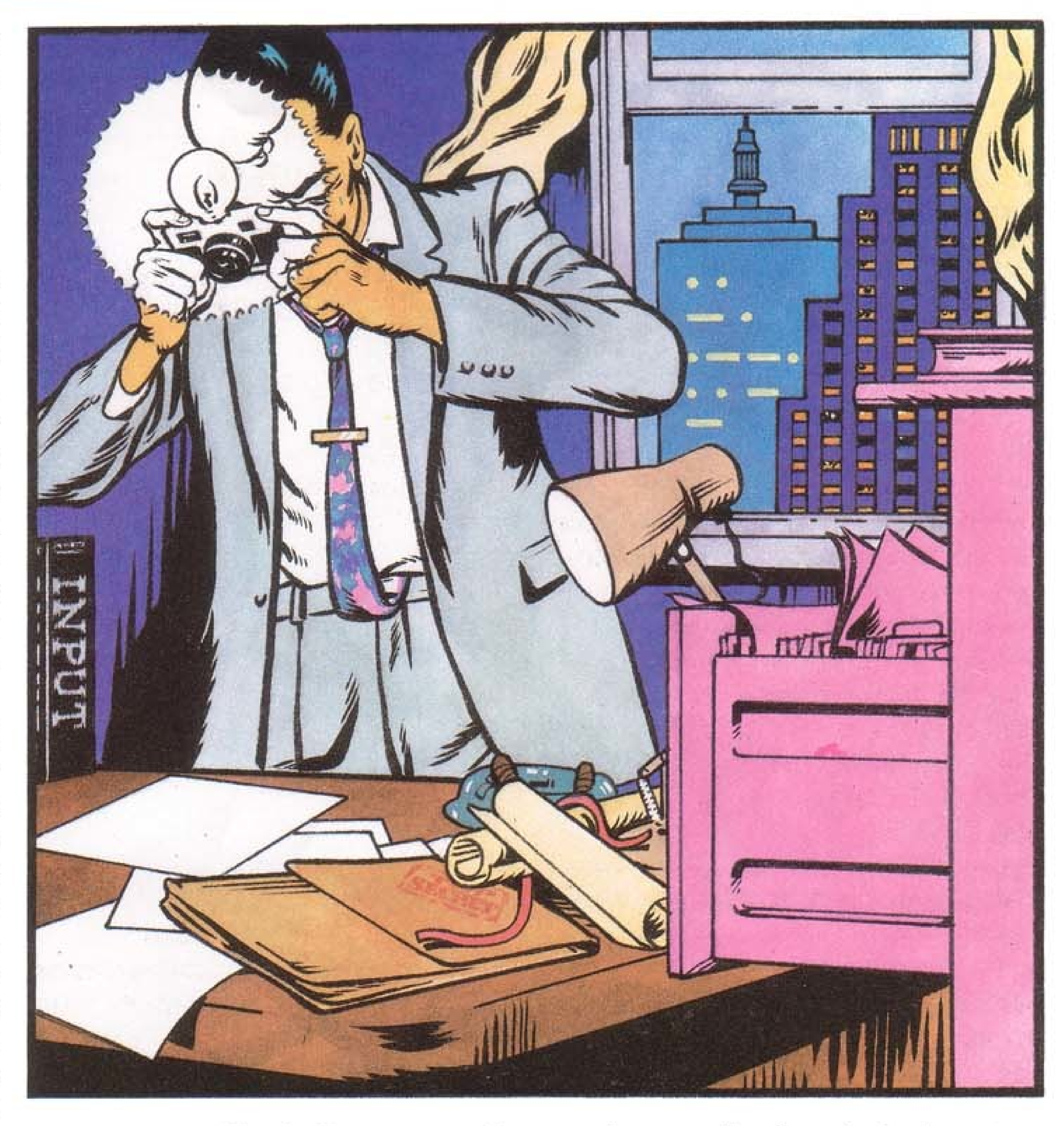
Todos los programas de LOGO se construyen de esta forma. La fragmentación de un programa en pequeños bloques constructivos hace más fácil su construcción y depuración.

En un próximo artículo veremos cómo realizar programas más complicados utilizando las técnicas aquí descritas.

ENVIA MENSAJES SECRETOS

- PRODUCE TUS PROPIOS CODIGOS SECRETOS
- CODIGOS DE DISTANCIA
- EL CIFRADO DE ST. CYR
- CODIGO MORSE

La codificación y el cifrado, que unen el mundo de la antigua Grecia con el mundo de la microelectrónica, son susceptibles de ser eficazmente manejados por los programas de ordenador.



La mayoría de la gente se sirve de los códigos en su vida diaria. Cuando alguien pide un tornillo del número 8 o un balón de fútbol del tamaño 5, está utilizando un código. Análogamente, las ecuaciones aparentemente incomprensibles de un científico nuclear representan la utilización de un código que permite expresar brevemente relaciones complicadas. Esto se podría hacer también utilizando palabras corrientes, pero, con frecuencia, requeriría mucho más trabajo.

En todos los anteriores ejemplos el interés reside en mejorar la comunicación de los datos, más que en buscar el secreto de la información; otra de las razones para el uso de códigos es ahorrar dinero o espacio de almacenamiento. Por ejemplo, una empresa puede redactar contratos que utilicen cláusulas y frases estándar. Si el almacenamiento se va a hacer en cinta o disco, se puede ahorrar una gran cantidad de valioso espacio codificando con números las secciones que se repiten más a menudo. A esto se le llama compresión de los datos.

Análogamente, las respuestas estándar utilizadas en juegos de aventuras, como "The Hobbit" o "Valhalla" se codifican para ahorrar espacio de almacenamiento.

Los griegos inventaron la ciencia de enviar mensajes secretos, de forma que no es sorprendente que el nombre utilizado normalmente para la codificación —criptografía— se derive de dos palabras griegas: *kryptos* (secreto) y *graphos* (escribir). Los términos **código** y **cifra** realmente tienen un significado ligeramente diferente, relacionado con las dos maneras en que pueden enviarse los mensajes. Cuando la información se transfiere letra por letra, se habla de cifrado. Si en cambio se transforman palabras completas o grupos de palabras en otras palabras o números utilizando un diccionario especial, a esto se le llama codificación. En la práctica, el término código se utiliza para referirse a códigos y cifras.

CODIGOS SECRETOS

La codificación y decodificación de mensajes secretos fue en otra época un área confinada principalmente a las actividades militares, o al menos a los Servicios de Inteligencia. Sin embargo hoy, el amplio uso de las líneas telefónicas públicas y de los canales de transmisión por satélite para transmisión de datos comerciales importantes, ha aumentado la necesidad de la codificación.

En los albores de la revolución informática, se utilizaban los comptómetros de **IBM** para romper los códigos de la Segunda Guerra Mundial. Desde entonces, todo nuevo avance en la tecnología de los ordenadores ha sido ansiosamente seguido por los maestros de espías y los

reventadores de códigos. Actualmente, con una buena programación y la cifra correcta, un ordenador doméstico puede funcionar como cualquier máquina convencional de criptografía. Este es el primero de dos artículos que mostrarán cómo utilizar el ordenador para producir mensajes secretos codificados, siguiendo varios métodos diferentes que, como el propio espionaje, se sitúan a niveles diferentes de sofisticación.

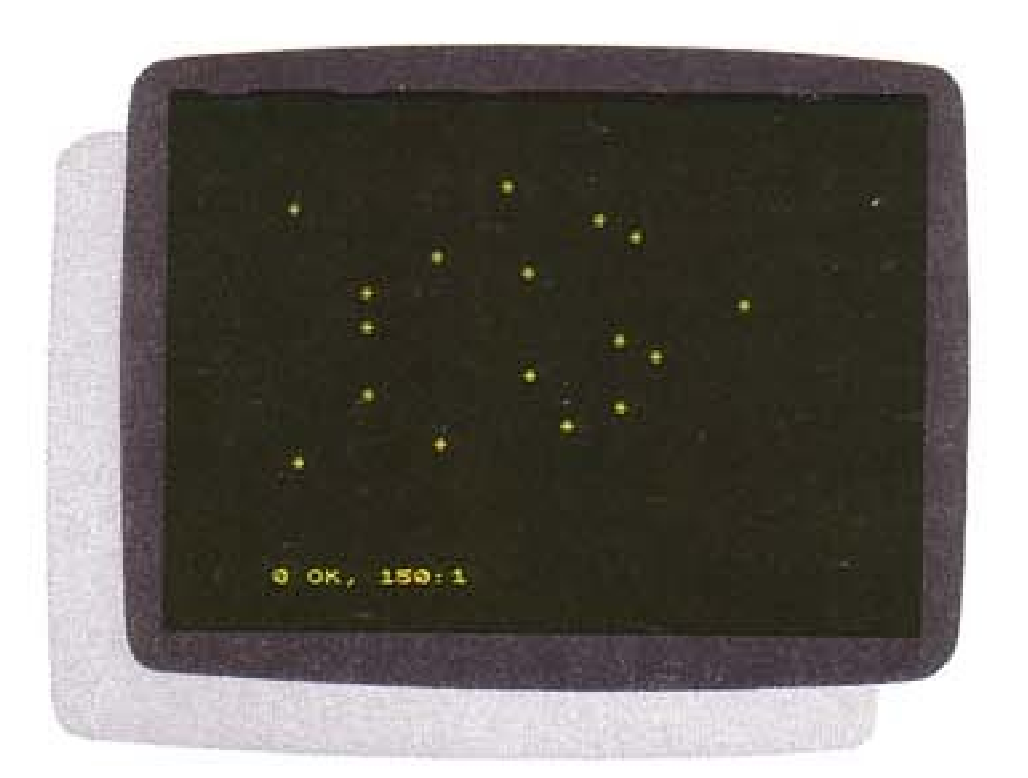
Aunque no seas un agente internacional, los métodos seguidos tienen interés por sí mismos, y siempre podrás servirte de ellos para enviar mensajes cifrados a otros amigos que tengan ordenador. De hecho, en este artículo hay un mensaje escondido en alguna parte.

CODIGOS DE DISTANCIA

La sucesión aparentemente aleatoria de símbolos mostrada en la foto de la pantalla es un ejemplo de este código. Como su nombre sugiere, se trata de un código que está basado en la distancia de un símbolo particular a un punto dado.

Este tipo de código fue utilizado hace más de 2000 años por el general griego **Lisandro.** Las distancias de las muescas a partir de la hebilla del cinturón de uno de sus esclavos componían un mensaje secreto que ayudó al general a derrotar al Imperio Persa.

Se puede formar sencillamente un código de distancia, poniendo las letras del alfabeto en la primera línea de un papel cuadriculado y componiendo el mensaje como se muestra en las figuras 1 y 2. Mientras se ve la clave de las letras en la línea superior, el mensaje es fácil de entender. Sin embargo, si se suprime esta clave, no es fácil decodificar la información.



Pantalla del código de distancia.

Para hacer el descifrado todavía más intrincado, haz una rotación de la clave de las letras en la primera línea de la página. Se puede, por ejemplo, empezar por la N y cuando se llega a la Z, simplemente empezar otra vez por la A. A esto se le llama rotación cíclica.

El programa funciona creando un bucle y sirviéndose de la facilidad de LEN para evaluar el valor equivalente ASCII de cada letra del texto en claro (líneas 90, 140). La mayoría de los ordenadores incorporan una función ASCII que permite representar las letras por medio de números. En todos ellos, excepto el ZX81, las letras toman los mismos valores.

Después de convertir el mensaje en una serie de números, es fácil codificarlo utilizando una transformación lineal inmediata. Por ejemplo, en el programa, la letra V se traduce al equivalente ASCII de V menos 26 (línea 130).

Sólo falta utilizar la función TAB para imprimir el asterisco a la distancia adecuada a partir del lado derecho de la pantalla (línea 120) y el proceso de codificación está completo.

Teclea

```
 20 BORDER 0: PAPER 0: INK 7:
    CLS
 30 PRINT TAB 8; "□Código
    distancia□": PRINT
 40 PRINT INK 2; PAPER 7;
    FLASH 1;AT
    6,10;"□□AVISO□□":
    PRINT
 50 PRINT "No dejar espacios
    entre palabras"
 60 PRINT : PRINT
 70 INPUT "Cuál es su
    mensaje>"'a$
 80 FOR i=1 TO 400: NEXT i:
    CLS
 90 FOR i=1 TO LEN (a$)
100 LET b$=a$(i)
110 LET v=CODE (b$)-96
120 IF v<=32 THEN PRINT TAB
    v; INK 6;"*": GO TO 150
130 LET v=v-26
140 PRINT TAB (v); INK 6;"*"
150 NEXT i
```

UTILIZACION DEL CODIGO

Aunque el código de distancia pueda parecer demasiado sencillo para ser eficaz, tiene algunos factores a su favor. En primer lugar, antes de poder descifrar con éxito un código, hay que darse cuenta de que realmente éste existe. Y como es muy fácil enmascarar una sucesión de puntos (o asteriscos) aparentemente aleatoria en un dibujo, por lo demás inocente, hay bastantes probabilidades de que un mensaje codificado de este tipo pase inadvertido.

Durante la última guerra mundial, los agentes alemanes utilizaron este



truco. Una inspección detenida a la inocente imagen de un jardín, reveló que las pinzas de la ropa en una cuerda de tender componían un mensaje secreto.

Una manera de hacer el código de distancia aún más difícil de romper es reestructurar el programa de forma que la clave de las letras sea realmente aleatoria. Un experto consciente de que tiene ante sí un código de distancia necesita ensayar, como mucho, 26 combinaciones antes de resolver el problema. Sin embargo, si el orden de las letras de la clave es aleatorio, el número de combinaciones posibles aumenta enormemente.

EL CIFRADO DE ST. CYR

Los romanos tomaron el relevo de los griegos como maestros en criptografía. **Julio César** inventó un método de cifra por sustitución directa en el que cada letra se sustituía por la que estaba situada en el alfabeto tres lugares a la derecha. Así la A se convierte en D, la B se convierte en E, etc. Al final del alfabeto, la X se convierte en A, la Y en B y la Z en C. Utilizando este método el mensaje VIENE EL ENEMIGO se convierte en ZLHPH HN HKHOLJR.

Como se verá más adelante, el código de **César** es un caso especial del llamado cifrado de **St. Cyr,** y el resultado anterior se puede comprobar fácilmente ejecutando el siguiente programa y utilizando como número clave 3333333.

Con la caída del Imperio Romano cesaron los desarrollos en criptografía, y a pesar del creciente uso de códigos en los siglos XVI y XVII, hasta el siglo XIX no se produjeron mejoras significativas del código de **Julio César,** realizadas por la academia militar de **St. Cyr.** El cifrado de **St. Cyr** es sorprendentemente sencillo. Se compone con tres alfabetos en una escala deslizante. El alfabeto de abajo es el del texto en claro y sus equivalentes en cifra se toman de la línea superior. Partiendo de un punto de origen predeterminado, la palabra INPUT se transforma en AFHML. Una de las ventajas del cifrado de **St. Cyr** es que cada letra se puede codificar utilizando un alfabeto equivalente distinto. Esto hace muy difícil reventar un código.

En el programa de cifrado de **St. Cyr,** se ha incorporado esta utilidad por medio de un número clave, para obtener una seguridad extra. Todo el que tenga acceso a un listado de programa no podrá resolver el código, a menos que conozca también los siete números secretos.



Teclea

```
20 BORDER 0: PAPER 0: INK 7:
   CLS
25 POKE 23658,8
30 PRINT TAB 10;"CIFRA DE
   ST.CYR": PRINT
40 PRINT INK 2; PAPER 7;
   FLASH 1;AT
   6,10;"□□AVISO□□"
50 PRINT "No dejar espacios
   entre palabras"
60 PRINT : PRINT
70 PRINT "PULSA 1 si deseas
   codificar"
80 PRINT "PULSA-1 si deseas
   decodificar"
90 INPUT s
100 INPUT "escribe tu
   mensaje"'a$
110 PAUSE 50: CLS
120 INPUT "escribe siete
   números"'n$
130 PAUSE 50: CLS
140 FOR k=1 TO LEN a$
150 LET l=k-INT (k/7)*7+1
160 LET t=CODE (a$(k))+
   (s*VAL (n$(l)))
170 IF t>90 OR t<65 THEN LET
   t=t-(s*26)
180 PRINT CHR$ (t);
190 NEXT k
```

Este programa opera de una manera muy parecida al programa de

códigos de distancia. Después de leer cada letra del texto en claro, se convierte primeramente en el equivalente numérico del carácter ASCII correspondiente. Luego se utiliza la función VAL para incrementar este número en una cantidad que depende del número deducido de su clave numérica (línea 160). Después de una comprobación para asegurarse de que el resultado está dentro de márgenes aceptables (línea 170), se aplica la utilidad CHR$ para sacar el mensaje codificado.

En los primeros días de junio de 1944 se podría haber codificado el siguiente mensaje: TROPASCAPTURADASENELPUENTEPEGASO. Utilizando el programa con el número clave 3821105, el texto codificado se convierte en: BTPQAXFIRUVRFGIUFOEQSCGOUEUHOCTP. Utilizando una variable indicadora S, que puede tomar los valores +1 ó −1, se puede utilizar el mismo programa para la decodificación. El número clave de siete cifras depende de la elección personal. Sin embargo, como estas cifras se confían frecuentemente a la memoria, es una buena práctica recurrir a algún dato familiar, tal como algún número de teléfono.

Como hay diez millones de valores diferentes para el número clave, el cifrado de **St. Cyr** es realmente de decodificación muy laboriosa. No obstante, se puede introducir una dificultad adicional codificando dos veces, es decir, poniendo el mensaje ya codificado como entrada al programa codificador por segunda vez.

Considérese el mensaje MARTINEZ ES UN ESPIA. Y utilizando los números clave 2501000 y 2506900, la secuencia cifrado-descifrado es la siguiente:

Texto	Número clave	
MARTINEZESUNESPIA		
	2501000	Codifica
RASTINGEETUNEUUIB		
	2506900	Codifica
WAYCINIJEZDNEWZIH		
	2506900	Decodifica
RASTINGEETUNEUUIB		
	2501000	Decodifica
MARTINEZESUNESPIA		

CODIGO MORSE

Incluso el más seguro de los códigos es poco útil si no se puede transmitir rápidamente. **Napoleón** hizo algunos intentos para resolver el problema de la velocidad construyendo torres de señales por toda Francia. Para pasar mensajes de una torre a otra se utilizaba un semáforo primitivo.

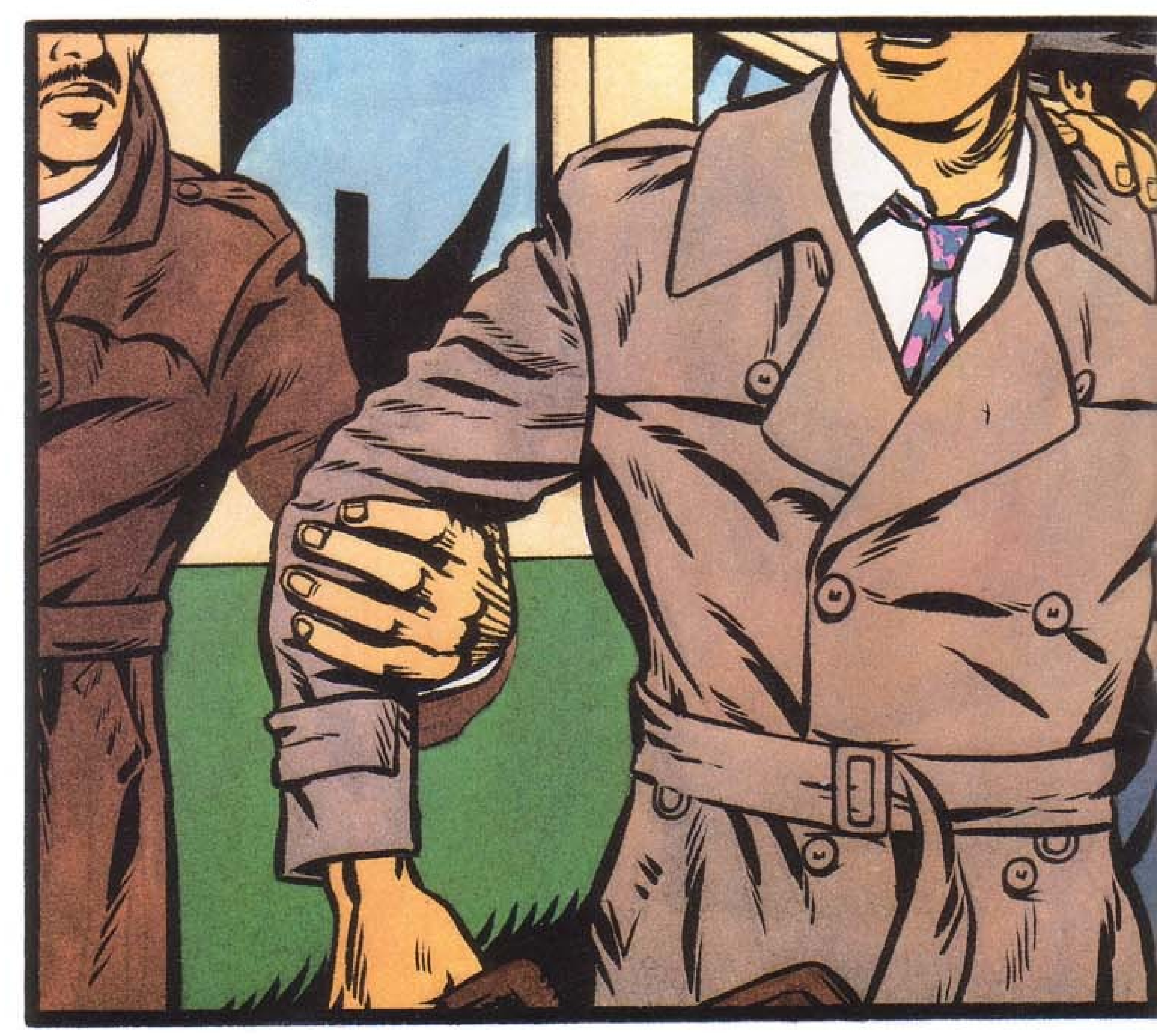
Sin embargo, lo que realmente aceleró la transmisión de mensajes secretos fue la invención del telégrafo eléctrico y la cifra de sustitución conocida por código Morse. El inventor americano **Morse** estableció un código en que las letras estaban sustituidas por puntos y rayas.

El programa del código Morse vale para cifrar texto, o decodificar una serie de puntos y rayas a sus letras adecuadas. Por ejemplo, la conocida señal de socorro SOS se convierte en /···/---/···/ y el texto ESCAPA A MEDIANOCHE pasa a ser /·/···/-·-·/·-/·-- /·-/·-/--/·/-··/··/·-/-·/---/-·-·/····/·/

Prueba los dos fragmentos de texto en el siguiente programa. Actuando en sentido inverso, si ejecutas el programa e introduces --0--0---000--0---***** aparecerá en la pantalla del televisor la palabra MAYDAY. Se utilizan los asteriscos para indicar al ordenador que se ha completado el mensaje.

En la primera parte de este programa se establece una cadena de caracteres con 130 puntos y rayas, para representar el equivalente Morse del alfabeto en orden secuencial (línea 250). Aunque el signo menos es un buen símbolo para representar una raya, hemos preferido utilizar un asterisco como símbolo gráfico para el punto, en lugar del punto ortográfico.

Se ha previsto un campo de cinco caracteres para cada letra, cuando bastaría con cuatro para todo el alfabeto. Se ha hecho esto para dar la oportunidad de ampliar el programa a quien desee introducir números que tengan códigos más largos. La sección de codificación es muy parecida a la de los primeros programas. Cada letra del texto es leída y con-



vertida en un número entre 1 y 26.

Al número equivalente se le aplica un factor de escala de cinco y la subcadena de puntos y rayas que resulta es lo que se imprime (línea 100).

Para la subrutina de descifrado que constituye la parte final del programa, se utiliza una técnica de búsqueda (línea 225).

Después de leer las señales Morse, el ordenador buscará en la serie hasta que encuentre una subserie idéntica. Cuando esto suceda, es inmediato transformar la posición de la serie en el correspondiente código numérico ASCII. Nuevamente se utiliza la función CHR$ para sacar el resultado requerido.

Teclea

```
10 BORDER 0: PAPER 0: INK 7:
   CLS : DIM a$(26,4): LET
   s$="": LET f$="□□□□□"
15 POKE 23658,8
20 FOR x=1 TO 26: READ
   a$(x): NEXT x
30 INPUT "Codificar(1) o
   Decodificar(2)";r
40 IF r=2 THEN GO TO 140
60 INPUT "MENSAJE PARA
   CODIFICAR"'m$
70 FOR x=1 TO LEN m$
80 IF m$(x)="□" THEN PRINT
   "□□□□";: GO TO 110
90 LET p$=m$(x)
100 PRINT "□□□□";a$((CODE
    p$)-64);
110 NEXT x
120 PRINT ''''TAB 10;"PULSA
    CUALQUIER TECLA PARA
    COMENZAR";: PAUSE 9999
130 RUN
140 INPUT "MENSAJE PARA
    DECODIFICAR";m$: LET
    m$=m$+"□"
160 FOR x=1 TO LEN m$
170 LET k$=m$(x)
180 IF k$="□" THEN GO TO 220
190 LET s$=s$+k$
200 NEXT x: GO TO 120
210 IF LEN s$>5 OR LEN s$ ^1
    THEN PRINT "ERROR": GO
    TO 120
220 IF LEN s$<>5 THEN LET
    s$=s$+f$(TO 5-LEN s$)
225 FOR h=1 TO 26: IF a$(h)=
    s$ THEN PRINT CHR$(h+64);
230 NEXT h
240 LET s$="": GO TO 200
250 DATA "0-","-000","-0-0",
    "-00","0","00-0","--0",
    "0000","00","0---",
    "-0-","0-00","--","-0",
    "---","0--0","--0-",
    "0-0","000","-","00-",
    "000-","0--","-00-",
    "-0--","--00"
```



Cifrado de St. Cyr. La pantalla muestra el mensaje y las dos opciones.

En un próximo artículo hablaremos de otros tipos de códigos.

PANORAMA PARA MATAR

Prácticamente agotados para el *software* los temas de marcianitos, las nuevas tendencias se centran cada vez más en las películas. El juego que examinamos a continuación recoge tres de las secuencias de acción de la película más reciente de **James Bond,** *«A view to a kill»* (Panorama para matar). Se presentó durante el pasado mes de junio en Londres, con el atractivo de las chicas **Bond** acudiendo al *stand* de la compañía que lo ha desarrollado. Ahora acaba de llegar a España.

Partiendo del guión original de la película se diseñaron tres juegos, que se traducen en otros tantos módulos perfectamente diferenciados. Probablemente el único punto en común de todos ellos sea la constante de la serie: **James Bond** nunca muere.

Como era de esperar, el juego comienza con una atractiva presentación en la que posa el agente «con licencia para matar» acompañado por su pistola.

A diferencia de otros programas, aquí no es el juego quien controla y domina al personaje sino, al revés, es la historia quien gira en torno a él.

Cada uno de los tres juegos se puede afrontar individualmente o se puede avanzar a través de ellos.

El primer módulo cobra vida en París. La malvada **Mayday** acaba de asesinar a un agente en la torre **Eiffel.** Perseguida por **Bond** salta en paracaídas. Tu misión (como **Bond** que eres) consiste en perseguirla por las calles de la capital francesa en un taxi robado, tratando de interceptarla en su punto de aterrizaje. En el laberinto de calles

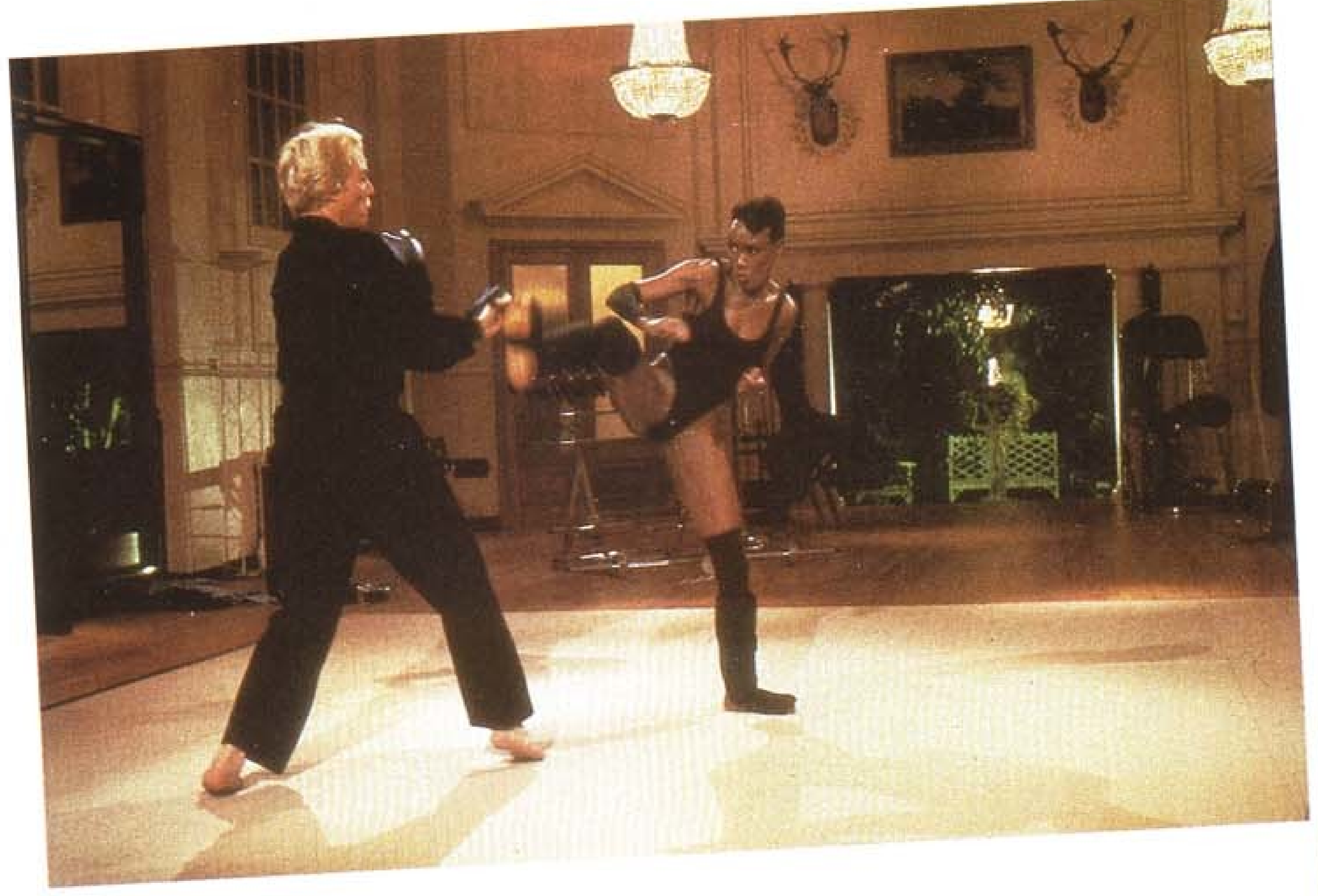


algunas son de dirección única y si tomas varias en dirección prohibida te perseguirá la policía. Si no logras atraparla, el juego se transforma en una persecución de coches por las calles de París.

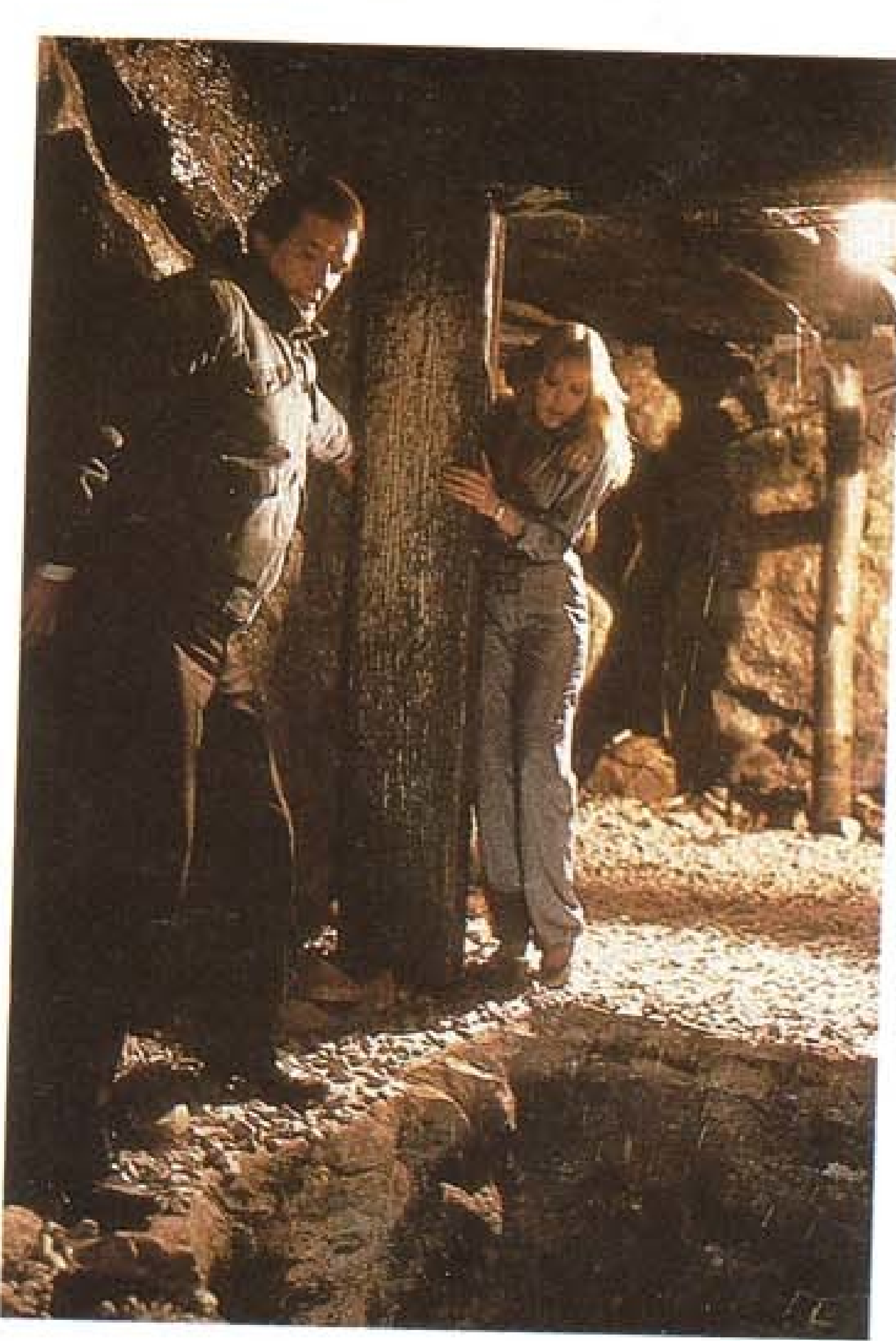
La pantalla se divide en dos áreas. Arriba una sensacional vista tridimensional, desde la ventana delantera del taxi, evoluciona a medida que avanzamos entre los edificios. La otra zona es una vista parcial del plano de la ciudad, en el que se puede apreciar el taxi y el paracaídas, deslizándose de manera que siempre podemos ver la posición que ocupamos. La hora es igualmente visualizada, pues el juego evoluciona en tiempo real. Igualmente, los límites de velocidad vienen impuestos. El control del automóvil no resulta fácil en un principio, pero un poco de práctica lo soluciona.

La segunda parte tiene por escenario el **City Hall** (Ayuntamiento) de San Francisco. Es un juego de aventuras en el que es acompañado por una preciosa chica, **Stacey.** Han sido capturados por el villano **Max Zorin,** que lanza una botella de líquido inflamable en el ascensor donde viajan obligadamente ambos. El indestructible agente logra escapar, pero su rubia acompañante queda atrapada en el edificio en llamas.

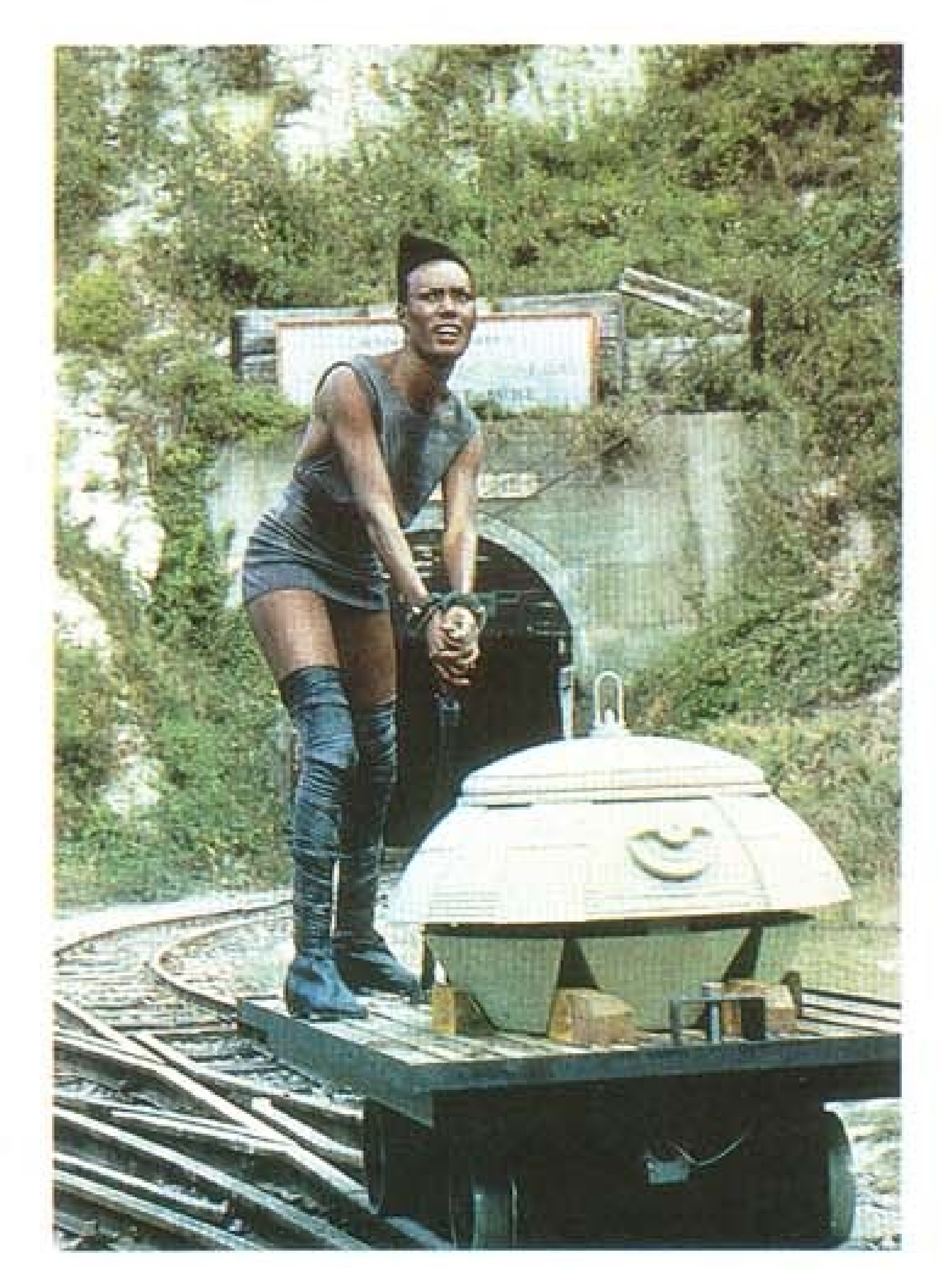
DATOS GENERALES	CALIFICACION
TITULO: A view to a kill	GRAFICOS: 3 sobre 5
FABRICANTE: Domark	COLOR: 4 sobre 5
ORDENADOR: Spectrum	PRESENTACION: 4,5 sobre 5
MEMORIA: 48 K	INTERES: 4 sobre 5
CLASE DE PROGRAMA: Aventura de espías	REALISMO: 4 sobre 5



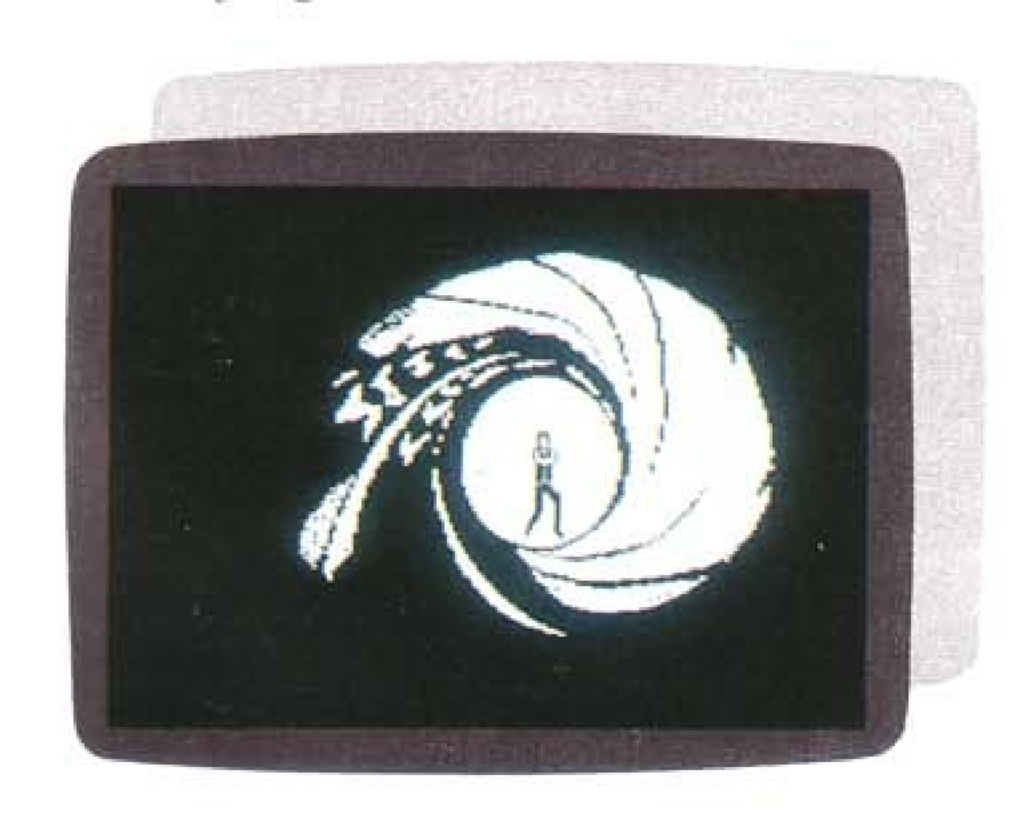
Algunas escenas comprometidas de «Panorama para matar» Cortesía de Cinema Internacional Copr.



El City Hall arde.



Esta pantalla da comienzo al reparto en el juego.



En nuestra evolución por las diversas salas podemos recoger objetos actuando con el *joystick* o el teclado sobre los letreros en los que se ofrecen los jegos de aventuras clásicos). Las salas están dibujadas en forma tridimensional y en la parte inferior podemos observar cómo es consumido el edificio por las llamas.
Como es natural, **Bond** logra escapar, porque conoce el camino de salida, pero es el jugador quien debe descubrirlo.
La tercera escena es tal vez la más peligrosa. Bajo el Valle del Silicio existe una mina secreta. **Zorin** ha amenazado con hacer estallar un paquete explosivo y **Stacey** y **Bond**



Stacey *es la principal compañera de* Bond *en este juego. Como es lógico, el agente la salva más de una vez.*

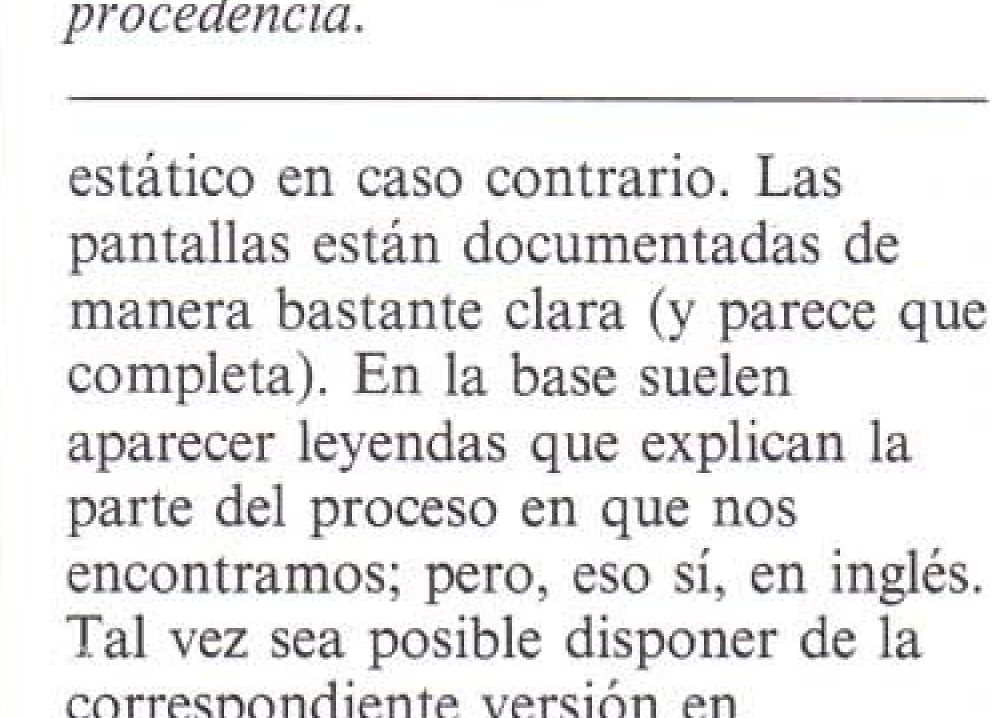


tendrán que desactivar el detonante. En caso contrario, se traducirá en la desaparición de San Francisco y Los Angeles, por la existencia de la conocida falla de San Andrés. Tienes que avanzar por la mina, pero con cuidado, pues las explosiones causan corrientes de agua. **Bond** cae en desgracia y **Stacey** logra escapar. La anteriormente enemiga **Mayday** es rescatada justo a tiempo para ayudar en el salvamento del agente británico. Nuevamente aparecen las opciones, **Bond** puede desactivar el detonador o llevárselo antes de que pase el tiempo límite.

EL CUERPO CIBERNETICO

Dos *cassettes,* un magnífico libreto y un póster a todo color forman un conjunto que explica de un modo espléndido los principales procesos que hacen que el cuerpo humano funcione. Está situado en el justo centro entre una lección de Biología y una película científica.
Siete programas, que se cargan por separado, describen desde la célula hasta cómo evolucionan las principales variables vitales durante un maratón.
El primer programa se titula Células y describe los diferentes tipos de ellas que hay, así como su composición y la manera en que las grasas, glucosa y aminoácidos penetran en ellas.
El proceso digestivo es el tema abordado por el siguiente programa. De modo dinámico aparece en la pantalla cómo van transformándose los alimentos ingeridos en productos químicos directamente utilizables por el cuerpo. Se puede elegir un tipo de alimento sean proteínas (carne, pescado, huevos), grasa (leche o mantquilla) o hidratos de carbono (patatas, azúcar) y examinar cómo evoluciona a través del sistema digestivo. Incluso está simulado el trabajo del hígado.
El sistema cardiorrespiratorio queda reflejado en el siguiente programa, que explica en un diagrama el circuito que recorre la sangre para ser oxigenada. Igualmente, son descritas minuciosamente las fases del ritmo cardíaco, cómo bombea el corazón la sangre y cómo pasa ésta a través de los pulmones. Gracias a las características dinámicas de la imagen se puede simular la respiración, analizando qué ocurre en los pulmones cuando corremos, aguantamos la respiración o existe hiperventilación.
Pero la descripción más completa del viaje que repite regularmente la sangre por las venas y arterias del cuerpo, se materializa en el programa destinado a la circulación. El diagrama recoge, con todo el detalle posible en la pantalla gráfica de un **Spectrum,** las principales zonas del cuerpo y describe, igualmente paso a paso, el avance del flujo sanguíneo.
Podemos ver esquematizados el corazón, los pulmones, el hígado, los riñones, el sistema digestivo, la cabeza y las extremidades superiores e inferiores del cuerpo.
Resulta interesante en todos los programas su interactividad; es decir, los programas no evolucionan a su aire, sino que el usuario puede elegir lo que desea conocer en un momento dado. Por otra parte, el programa avanza a medida que el usuario presiona una tecla (normalmente Space), por lo que la pantalla requiere continúa atención, bajo riesgo de permanecer estático en caso contrario. Las pantallas están documentadas de manera bastante clara (y parece que completa). En la base suelen aparecer leyendas que explican la parte del proceso en que nos encontramos; pero, eso sí, en inglés. Tal vez sea posible disponer de la correspondiente versión en castellano no tardando mucho.
El funcionamiento de los nervios es otro programa del conjunto. Muestra cómo éstos envían impulsos al cerebro y entrando en mayor detalle para describir cómo trabajan la vista y el oído en conexión con el mismo.
La detallada exposición del

DATOS PRINCIPALES	CALIFICACION
TITULO: Body Works	GRAFICOS: 5 sobre 5
FABRICANTE: Génesis	COLOR: 4,5 sobre 5
ORDENADOR: Spectrum	PRESENTACION: 5 sobre 5
MEMORIA: 48 K	INTERES: 5 sobre 5
CLASE DE PROGRAMA: Simulación del cuerpo humano	REALISMO: 5 sobre 5

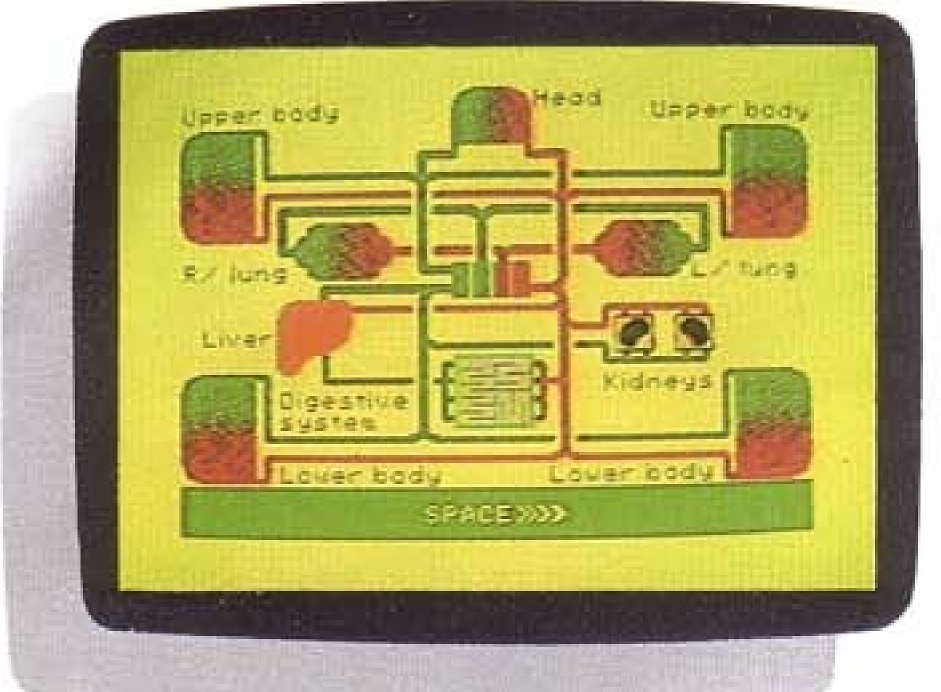


El cerebro reacciona a los estímulos nerviosos analizando rápidamente su procedencia.



Este diagrama permite seguir el funcionamiento de las válvulas cardíacas y los pulmones.

funcionamiento de los músculos merecería un capítulo aparte, pero baste decir que el programa correspondiente comienza describiendo cómo se contraen los músculos a nivel microscópico, pasando al detalle del control ejercido desde el cerebro.
Para que las ideas queden más claras, podemos controlar el funcionamiento esquemático de un brazo y observar qué ocurre con los principales pares de músculos extensores y flexores que participan en el movimiento. Para terminar, un programa sencillamente superior: el maratón dinámico. Podemos trazar, desde un menú, el perfil del corredor (podríamos ser nosotros), estableciendo la edad, condición de fumador, no estar en forma, etc., y elegir la distancia a recorrer. Obtenemos una visión áerea del atleta, el tiempo transcurrido y el electrocardiograma que muestra el ritmo cardíaco instantáneo. Esta última posibilidad puede ser sustituida a voluntad por una vista del perfil del terreno y el corredor avanzado por las cuestas y terraplenes. La parte principal de la pantalla es ocupada por los diagramas donde vemos la evolución de las variables vitales: cantidad de oxígeno respirado, volumen de sangre bombeado, velocidad de la respiración, temperatura, ritmo cardíaco, lactato, etc.
La velocidad de nuestro corredor puede ser controlada a voluntad, aumentando o disminuyendo desde el teclado. La representación es tan realista, que podemos fatigarle, incluso si no disponía de suficiente preparación podríamos forzarle al ataque cardíaco e incluso al colapso. Resulta enormemente interesante ver cómo evolucionan todas estas variables a medida que el deportista sube una pendiente o disminuimos su velocidad. Este paquete es uno de los más completos que hemos visto para el **Spectrum.** Su capacidad didáctica queda fuera de toda duda, por el factor dinámico de análisis que introduce. Sin embargo los programas no guardan un grado de complejidad uniforme, unos parecen totalmente adecuados para los más jóvenes, mientras que otros parecen destinados a personas con mayor grado de formación. Resta decir que el desarrollo estuvo a cargo de dos conocedores de la materia (un fisiólogo y un doctor en medicina) y, por otro lado, un diseñador gráfico.



Cabeza, tronco y extremidades son continuamente regidos por el flujo sanguíneo.

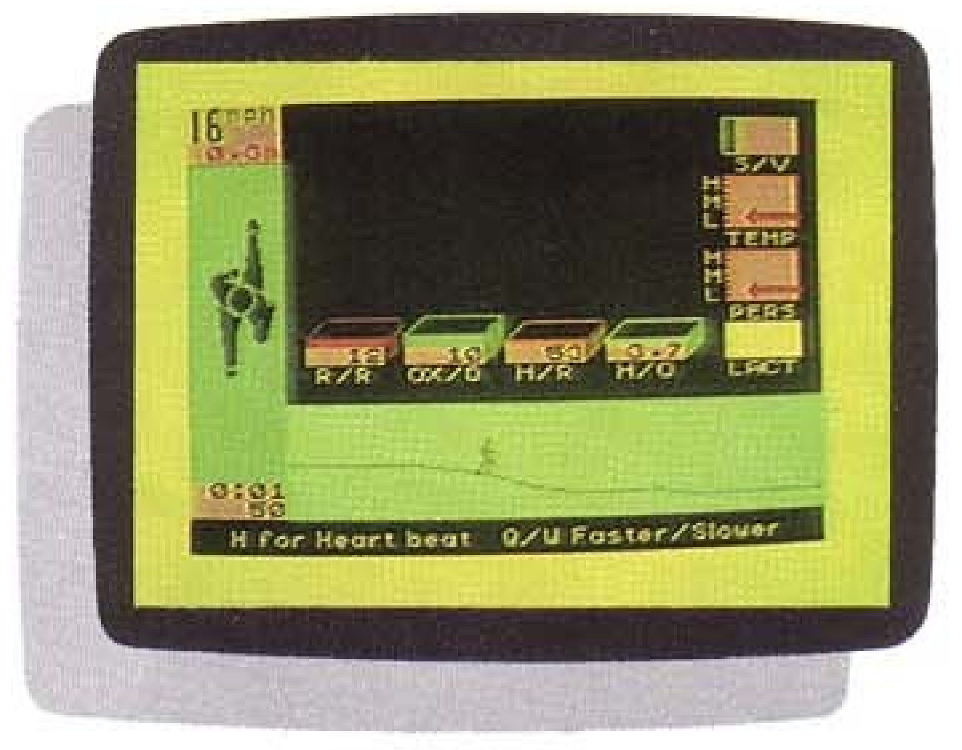
Simulación de las formas de onda producidas por un sonido puro en el aire.



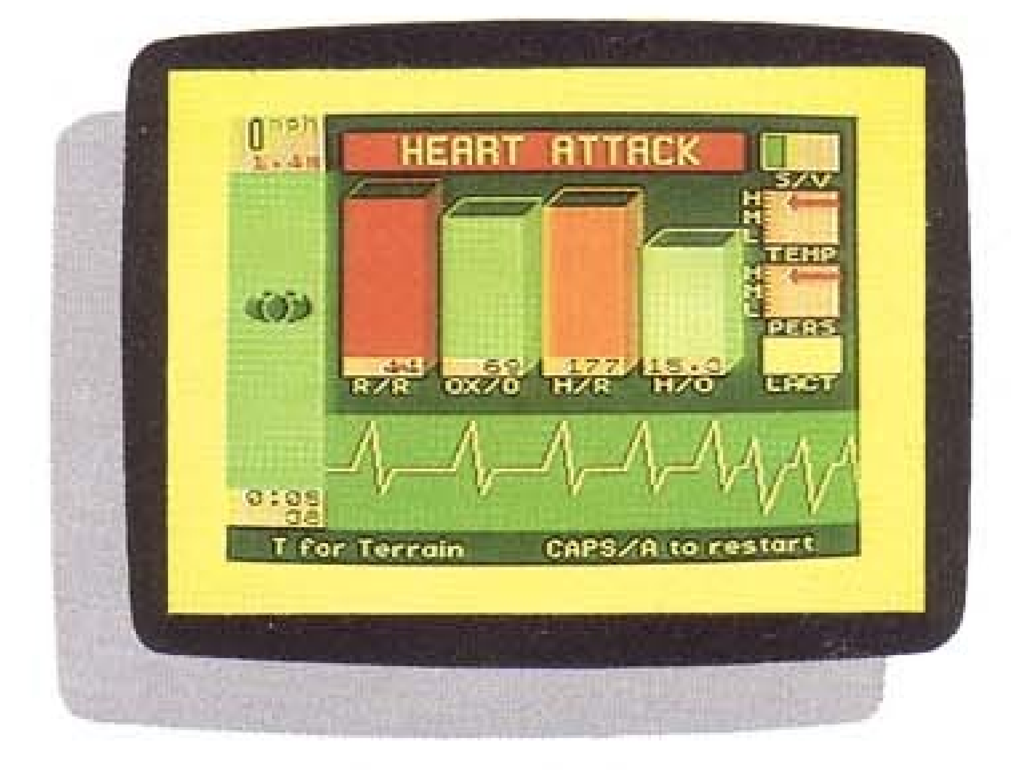
Desde el teclado se controlan los pares de músculos flexores/extensores.



Menú de elecciones para que tracemos el perfil de nuestro corredor.



Nuestro *atleta parte a buen ritmo, todavía sus constantes permanecen normales.*



Hemos forzado excesivamente el ritmo. La respuesta ha sido un ataque cardíaco.

EL COLEGIO DE LOS LIOS

Tal vez siempre hayas sido un estudiante ejemplar, pero ahora tienes la oportunidad de convertirte en un travieso colegial que protagoniza una arriesgada aventura, con la que probablemente habrás soñado alguna vez. ¿No resulta tentadora la idea de rescatar tu expediente y rectificar todas las notas que harán que seas inexcusablemente expulsado del centro docente? Pero, según siempre se comentó, los ansiados papeles se encuentran a buen recaudo, depositados en el interior de la caja fuerte que pusieron en la sala de profesores. ¡No importa, hay que conseguirlo!

El juego, a pesar de lo que puedas creer, es absolutamente realista. Incluso puedes escribir los nombres de tus propios compañero. No solamente tienen cabida Martínez, Gutiérrez o Corrales, los *«profes del cole»* también figurarán. El empollón, el matón y el llorica te acompañan en este colegio.

El chuleta que te roba el bocata de la cartera cuando te descuidas recibirá un certero puñetazo en el ojo, para que aprenda a no molestar. El profesor de Ciencias, que siempre te suspende, será el blanco de tu tirachinas.

Sin embargo, conseguir el expediente, que hace que los azotes en tus posaderas dependan de un hilo, no será tan fácil de conseguir. Como primer paso debes reunir todos los escudos que hay repartidos por la clase y los pasillos, en algunos casos es preciso emplear una sútil estrategia.

Pasemos revista a tus armas. Dispones de un tirachinas, tus puños y la puntería, que ha de ser muy fina para llegar a los escudos. Si derribas al *profe* con el primer *chinazo,* el segundo rebotará en su cabeza hasta alcanzar al escudo situado sobre él. Cuando le pegas un puñetazo a un *compi,* te subes encina de él y saltas hacia el escudo. Tendrás que ser rápido y actuar antes de que se recupere.

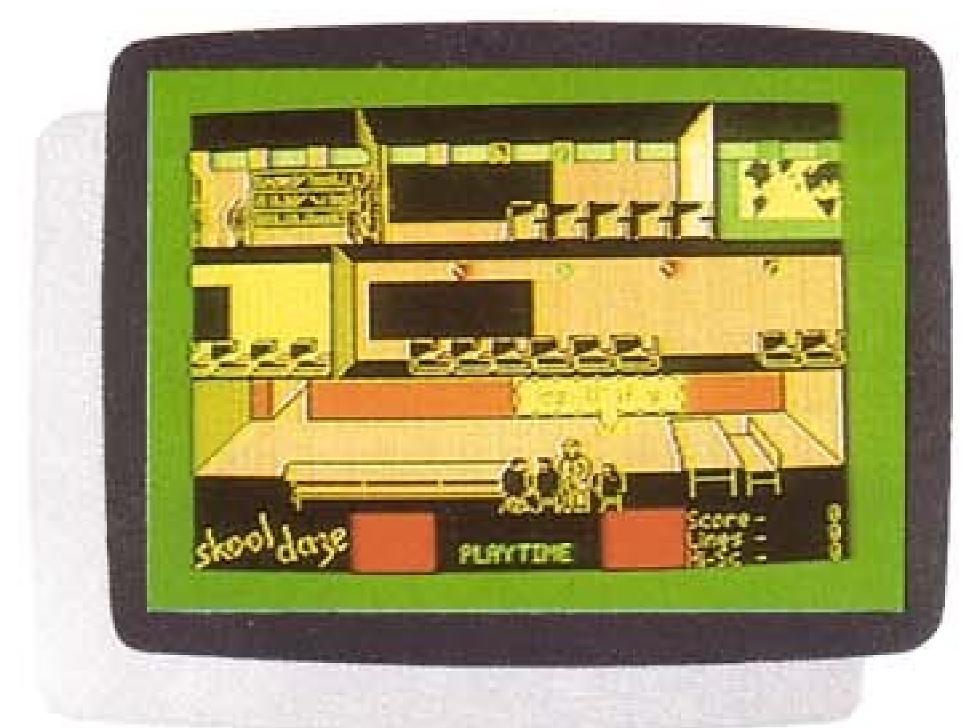
Para mantener tu coartada debes dar muestras de buen comportamiento. Has de parecer disciplinado y asistir a las clases, no escabullirte durante los recreos y no te sentarás en el suelo. Cuando quieras salirte de las normas procura no ser visto, si no los profesores te castigarán a copiar líneas.

Pero, descubramos cómo es el colegio. Consta de una planta baja, compuesta por el patio, el comedor y el gimnasio. El primer piso tiene la sala de profesores, una escalera, el aula blanca, el aula de exámenes y otra escalera.

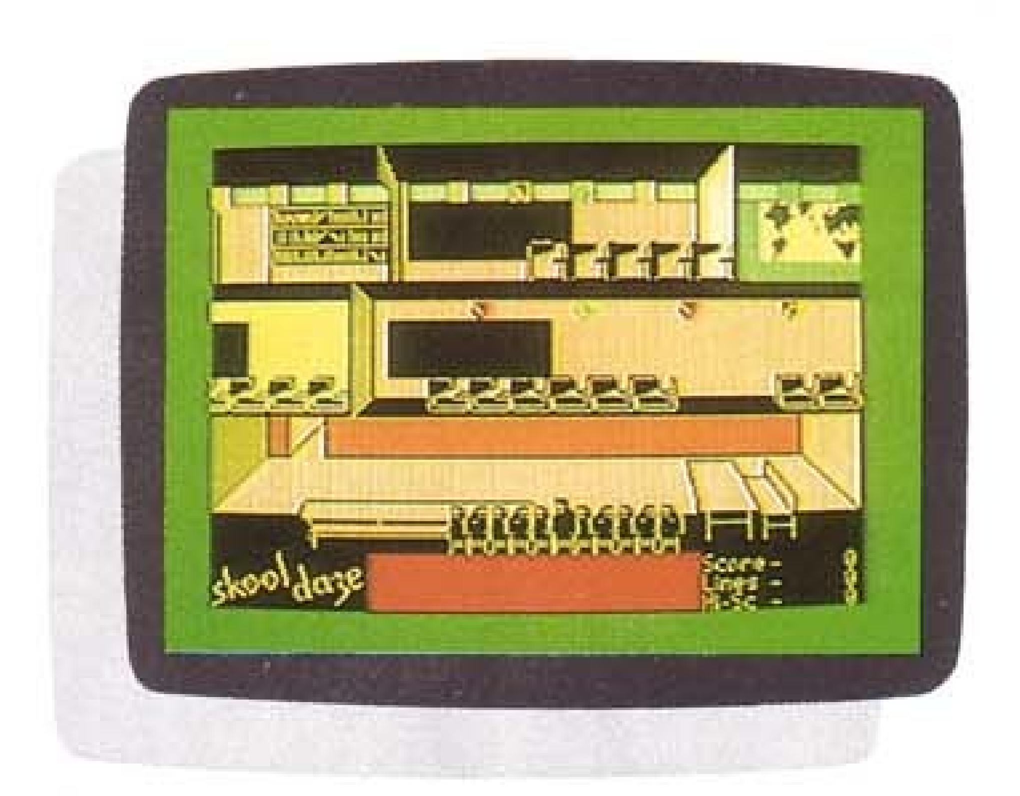
La segunda, y última planta, queda compuesta por el despacho del *dire,* una escalera, la biblioteca, la sala de lectura, el aula de Geografía, otra escalera y la terraza. Todo ello está representado en forma de vista tridimensional y simultáneamente se puede ver solamente un tercio de la superficie de las plantas.

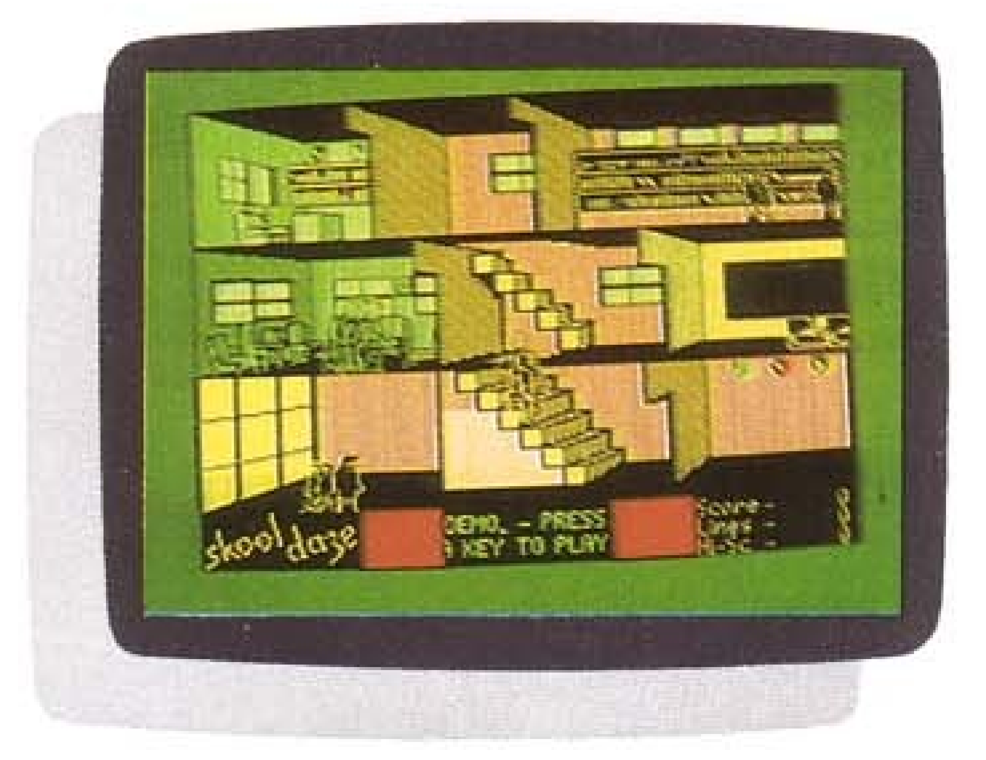


Pantalla de presentación del juego. El profesor siempre vigila a sus alumnos.



En estos recintos transcurre la historia donde te la juegas. Máxima precaución.





Las complicaciones tampoco faltan. Debes andarte con cuidado, porque si un condiscípulo comete una fechoría y te encuentran cerca del lugar de la acción, te culparán a ti. Algunas aulas están superutilizadas y hay menos pupitres que alumnos. Procura andarte listo, pues si el *profe* te pilla sentado en el suelo ya sabes... castigo y a copiar líneas. Los chivatos y las contagiosas paperas pueden significar castigos o ser enviado a casa una temporadita.

Algún consejo. Si te sientas en el último pupitre de la sala de lectura no tendrás problemas de sitio. Los escudos más próximos a las escaleras pueden ser alcanzados desde ellas, situándote a su altura. Cuando derribes a un *profe* y no puedas intentar coger el escudo, sal huyendo para evitar el castigo. Conseguirás más puntos si procuras

hacer la travesura cuando esté próximo otro *compi*. Le culparán a él sin duda.
Si te vieras obligado a escribir más de 1.000 líneas será el director en persona quien se encargue de expulsarte. Sin embargo, un punto débil. Si te encuentras en el aula blanca cuando rebasas las 1.000 líneas, el *profe* te sigue castigando a escribir más y más y habrá que desconectar el **Spectrum** para salir de este bucle sin fin.

DATOS GENERALES	CALIFICACION
TITULO: Skooldaze	GRAFICOS: 4,5 sobre 5
FABRICANTE: Microsphere	COLOR: 3 sobre 5
ORDENADOR: Spectrum	PRESENTACION: 4,5 sobre 5
MEMORIA: 48 K	INTERES: 5 sobre 5
CLASE DE PROGRAMA: Rebeldía escolar	REALISMO: 4,5 sobre 5

****

UNA TARDE EN LAS CARRERAS

Incluso si en las proximidades de tu casa no existe un hipódromo, podrás pasar toda una tarde en las carreras. Este juego parece haber sido diseñado para promocionar, aún más si cabe, el **Grand National** británico. La más prestigiosa competición ecuestre de aquel país. Su dureza está reconocida a nivel mundial. Los obstáculos contribuyen a que solamente pueda ser concluida por los mejores o los más afortunados. Pueden participar hasta cuarenta caballos, cabalgados por *jockeys* profesionales o por intrépidos aficionados, que ven así cumplido el sueño de su vida.
El juego, avalado por el organizador oficial de la competición, se divide en tres partes. En la primera aparece un menú, en forma de páginas, en el que podemos examinar la información relativa a los cuarenta caballos, visualizándose la referente a diez por cada una. Podemos obtener su calificación relativa, reducida a un número entre cero y diez, por contra a la utilizada en la realidad que es más completa. Sin embargo esta cifra nos orienta en cuanto a la forma en que se encuentra el equino.
Igualmente podemos hacer nuestras apuestas hasta un número de cinco. Elegimos el caballo y la apuesta que deseamos hacer por él. Una pequeña sofisticación del juego permite que sean directamente deducido el 10 por 100 correspondiente a Hacienda.
Por último podemos incluso ser jinete. Podemos elegir uno entre diez caballos de los cuarenta totales, que aparecen indicados con su número en video inverso.
Para ayudarnos en la elección debemos hacer uso de la información, personalizada para cada uno, que muestra la pantalla. Aparte de su número asociado, disponemos de datos tales como el nombre del animal, su estilo (los hay buenos *sprinters* y otros que hacen buenos tiempos medios) y el tipo de terreno que les es más apropiado, desde blando con barro y arena, hasta duro y tal vez pedregoso.



Hasta cuarenta caballos pueden participar en este prestigioso evento equino. Solo uno puede ganar.

La segunda parte es la carrera en sí. Aquí vemos al caballo desde arriba y también lateralmente. En la vista aérea aparecen también los caballos que se aproximan.
Haber elegido el caballo correcto es importante, pero no lo es menos la cabalgadura que llevemos a cabo. Se puede emplear el *joystick* o las teclas Z, O, X o P. Con ellas hacemos que el caballo se desplace

DATOS GENERALES	CALIFICACION
TITULO: Grand National	GRAFICOS: 3,5 sobre 5
FABRICANTE: Elite	COLOR: 3 sobre 5
ORDENADOR: Spectrum	PRESENTACION: 4,5 sobre 5
MEMORIA: 48 K	INTERES: 4,5 sobre 5
CLASE DE PROGRAMA: Simulación	REALISMO: 5 sobre 5



Por la recta de tribunas avanza el favorito. No conviene ser confiado, todos quieren ganar.

de izquierda a derecha y viceversa, empleamos la fusta para azuzarle y saltamos al llegar a los obstáculos. Una serie de indicadores nos mantiene al corriente del estado de la carrera. Así podemos conocer el número de caballos caídos, los retirados y los que se mantienen en carrera. La energía que empleamos, la velocidad y la posición son datos que van siendo actualizados. Asimismo podemos conocer el tiempo de la carrera, el número del siguiente obstáculo, el caballo que va en cabeza y la posición que tenemos en la pista. De acuerdo al estilo del caballo, debemos ser consecuentes en nuestra forma de cabalgar. A un caballo buen *sprinter* no podemos quemarle haciendo que lleve una carrera fuerte, a riesgo de agotarle innecesariamente, lo que le inutilizará para ganar en la recta final.
La última parte es el informe de la carrera. En la pantalla aparecen visualizados los datos pertinentes. Animo, cuarenta mil libras esterlinas esperan al jinete ganador y un buen montón de alfalfa de primera al caballo que montó.



La competición es dura y las apuestas pueden hacerte millonario si hiciste una buena elección.

****

CON UN PORSCHE Y A LO LOCO

Aunque la hoja de explicaciones no lo menciona por ningún lado, el 911 es un modelo de la marca **Porsche.** Sin embargo queda claro que **Dunlop,** fabricante de neumáticos entre otros, ha prestado su anagrama y asistencia técnica. En realidad el programa ha sido desarrollado primordialmente para su empleo como reclamo publicitario de **Dunlop.**
Con las 2.000 libras esterlinas que tenemos a nuestra disposición en principio debemos equipar a nuestro deportivo con los accesorios más adecuados para ganar un duro *rally*. De su correcta eleción depende el éxito y, probablemente, que no dejemos la vida en el intento y el vehículo hecho migas. Las cosas se ponen aún más difíciles. Nuestro rival es un tipo poco deportivo y sin escrúpulos, que pondrá el máximo empeño en sacarnos fuera de la carretera.
Cuatro menús permiten que controlemos nuestra inversión. El primero ofrece, como es lógico, la elección de los neumáticos (¡**Dunlop,** por supuesto!). Cinco son las posibilidades, desde los específicos para barro y nieve, hasta los de *rally* o superdeportivos, pero también podemos decidir que sean los estándar quienes calcen al 911.
La segunda elección afecta al combustible, elegimos entre cuatro tipos cuya clasificación nos es poco familiar: 2, 3, 4 estrellas o de alto octanaje.
La última es más diversa, podemos optar entre una dirección fuerte, suspensión dura o frenos potentes. De cualquier modo, el equipo extra se utiliza en las etapas impares que componen el recorrido total.
Sentados al volante, sea el teclado o el *joystick,* debemos intentar terminar con bien las ocho etapas, que transcurren a través de cuatro tipos diferentes de paisaje.
La pantalla de trabajo se divide por áreas. La fundamental es una vista aérea de un tramo de la carretera por el que avanzamos y en la que aparecen los diversos objetos que

DATOS	CALIFICACION
TITULO: 911 TS	GRAFICOS: 3 sobre 5
FABRICANTE: Elite	COLOR: 3 sobre 5
ORDENADOR: Spectrum	PRESENTACION: 4 sobre 5
MEMORIA: 46 K	INTERES: 4,5 sobre 5
CLASE DE PROGRAMA: Aventura automovilística	REALISMO: 3,5 sobre 5

debemos esquivar. En la parte inferior, ocupando una franja longitudinal está visualizado un mapa del recorrido total, quedando perfectamente delimitadas las ocho etapas, la posición que ocupamos en cada momento y la de nuestro contendiente.
A la izquierda tenemos el indicador de velocidad en sentido longitudinal y en el otro extremo de la pantalla otro indicador monitoriza la cantidad de daños que va acumulando nuestro vehículo.
En la parte superior están el logotipo de **Dunlop** y el anagrama identificador de este juego. Inmediatamente a su lado, ocupando la otra mitad superior de la



Este es uno de los cuatro menús de elección. En este caso hay cuatro variantes de neumático.

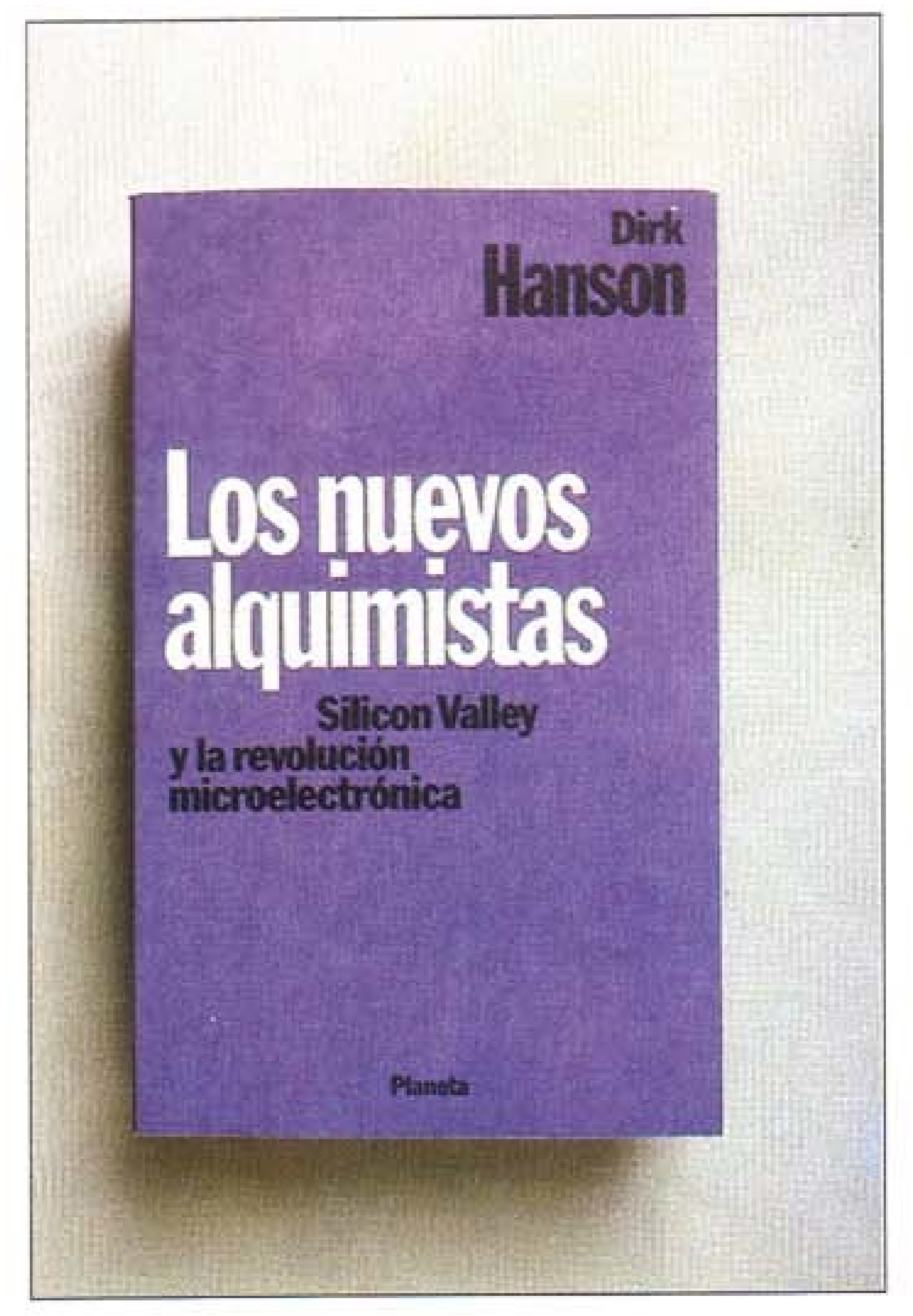


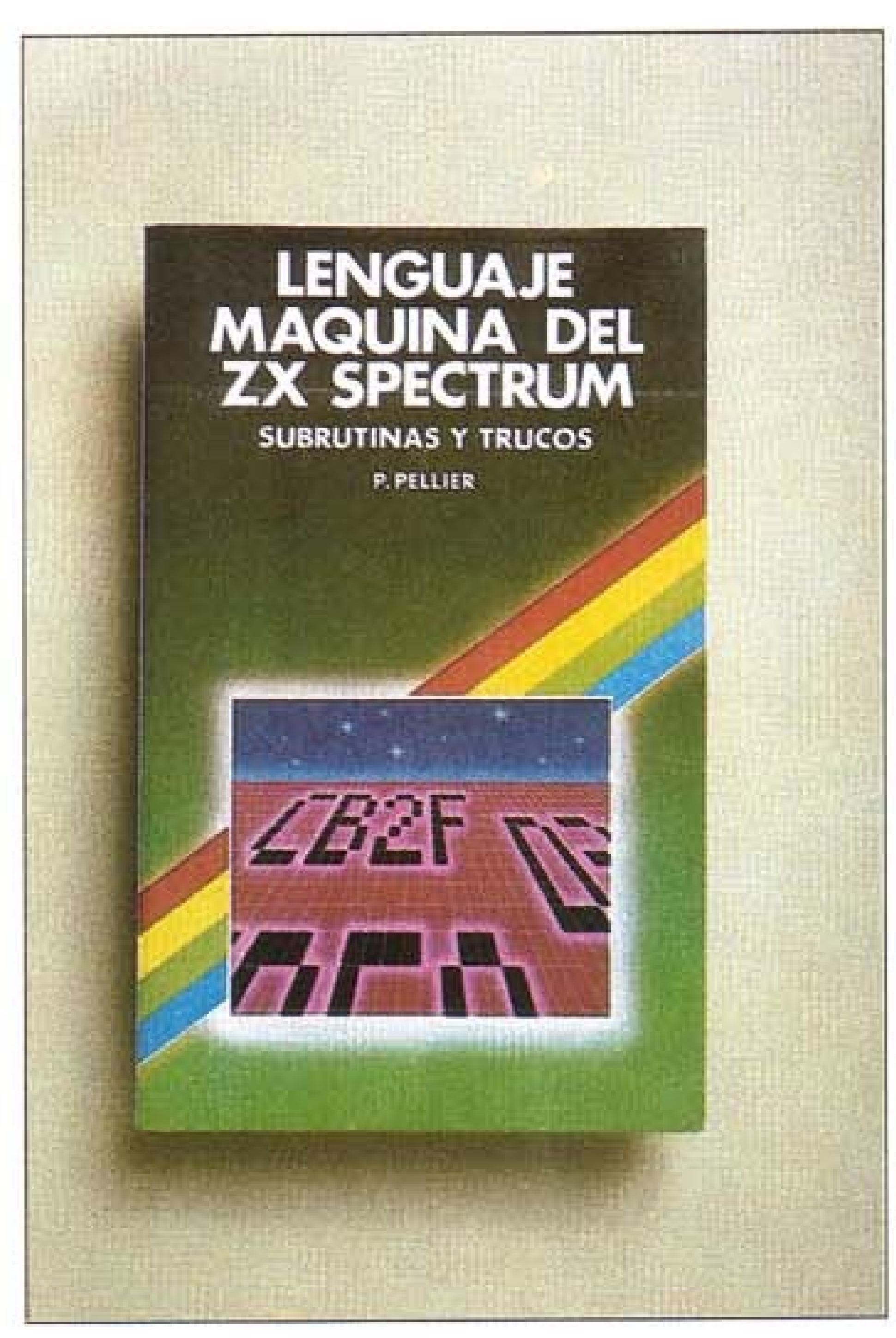
El 911 en solitario por un tramo poco difícil.

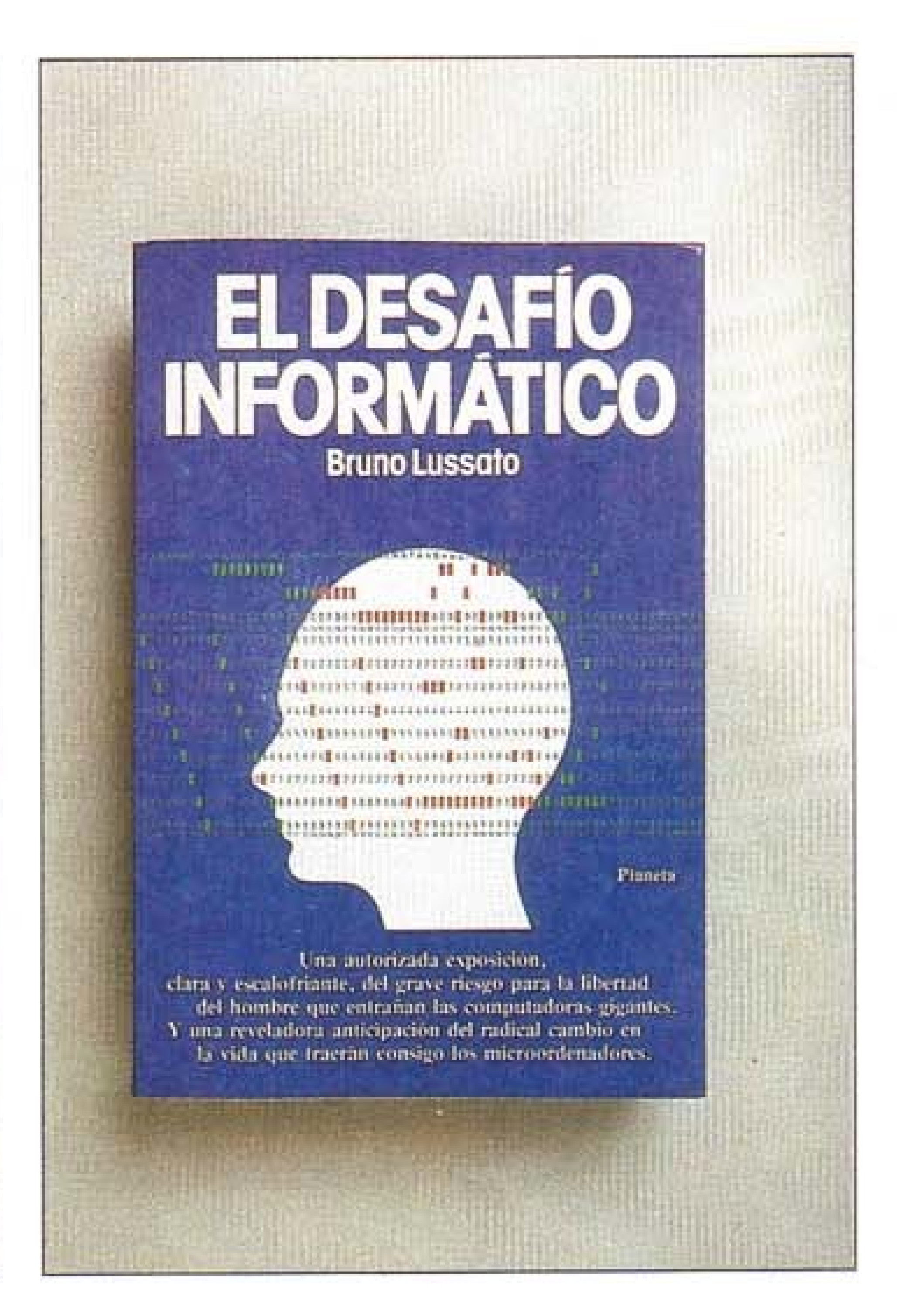


Nuestro contrincante es un corredor duro y sin escrúpulos, que no dudará en arrojarnos a la cuneta.

pantalla observamos diversos indicadores que son: la distancia que resta para finalizar la etapa, la puntuación lograda y el tiempo del que disponemos para llegar.
En las etapas impares podemos recoger el equipo, que en su momento compramos, y sacarle mayor rendimiento al coche.
¡Bruumm!







LOS NUEVOS ALQUIMISTAS

Autor: Dirk Hanson
Editor: Planeta
Páginas: 292
Precio: 800

Los nuevos alquimistas es una narración amena y detallada de la evolución de la tecnología electrónica desde los tiempos en que Edison fabricó su lámpara de incandescencia hasta nuestros días, repasando con especial detenimiento las circunstancias que propiciaron la aparición y el desarrollo, en Silicon Valley (Valle del Silicio), California, de la industria de los semiconductores. Es la historia de los hombres que consiguieron transmutar el silicio en oro o, lo que es lo mismo, la tecnología en un negocio en el que se invierten y se ganan —o se pierden— millones y millones de dólares.

Dirk Hanson relata el rápido progreso de la electrónica describiendo sus hitos: los primeros ordenadores, el transistor, los circuitos integrados, los microprocesadores, ordenadores personales, etc. Pero también cuenta las circunstancias que impulsaron y dieron forma a este desarrollo. Así, la importante influencia de los militares, que aportaron tanto su avidez de tecnología como sus tentadores presupuestos.

Analiza, además, las implicaciones y consecuencias que el vertiginoso progreso electrónico está teniendo en nuestra sociedad, e intenta hacer una prospección de lo que será en el futuro: la sociedad automatizada.

LENGUAJE MAQUINA DEL ZX SPECTRUM

Autor: P. Pellier
Editor: Gustavo Gili
Páginas: 126
Precio: 1.200

Este título es traducción de una obra originalmente escrita en francés. Se trata de una descripción del código máquina empleado por el microprocesador **Z80,** que incorpora el **Spectrum,** que comienza con la descripción del mapa de memoria, configuración del sistema y los clásicos registros internos del **Z80.**

La procedencia francesa del libro queda patente a lo largo de toda la obra. Por ejemplo, para el byte se emplea la traducción literal de ese idioma, aludiéndole como octeto. (¿No recuerda a un conjunto músico-vocal de ocho componentes?) y las memorias RAM y ROM responden a memoria viva y memoria muerta respectivamente.

En el capítulo segundo se hace necesario disponer de herramientas de programación básicas para sacarle el mayor partido a su lectura.

El capítulo tercero propone diversas subrutinas de interés general, que podrían ir incorporadas en nuestros propios programas, como es la multiplicación, la división o la generación de números aleatorios.

En resumen, se trata de una obra breve pero de utilidad práctica para quienes desean ampliar sus conocimientos en este campo a través de útiles rutinas de aplicación inmediata.

EL DESAFIO INFORMATICO

Autor: Bruno Lussato
Editor: Planeta
Páginas: 202
Precio: 700

El ordenador se plantea como un reto que todos deberemos afrontar en el futuro. Sin embargo, los agoreros se anticipan a presentarlo con una gran dosis de incertidumbre para el pobre mortal. A su entender los sistemas informáticos comienzan a significar una seria amenaza.

Bruno Lussato, el autor de este libro, parte de un supuesto diametralmente opuesto, y su intención es arrojar luz sobre el tema de la forma más coloquial posible.

El miedo desaparece con la información, parece ser la moraleja escondida tras **El desafío informático.** A lo largo de las páginas condimenta las peculiaridades y evolución de la industria informática con alusiones a la literatura y otras artes más conocidas por todos.

Esta obra desprende, casi de inmediato, la enorme fe que su autor tiene en el microordenador, como algo necesario para hacer que evolucione la nueva sociedad.

Fundamentalmente estamos ante una obra destinada a quienes desean conocer más a fondo los recursos ofrecidos por el ordenador. No se puede adjudicar a su contenido el calificativo de técnico, pero sí divulgativo.



SENCILLO, ASEQUIBLE, PROFESIONAL

ASI ES EL QL DE SINCLAIR, HECHO PARA NOSOTROS

Para los profesionales que necesitamos un teclado en nuestro idioma, QL nos ofrece, en castellano, su QWERTY standard de 65 teclas móviles.

Para los que deseamos comunicarnos a gran velocidad y capacidad con nuestro ordenador, QL nos presenta su lenguaje SUPER BASIC.

Para los que necesitamos gran margen operativo, ahora disponemos de un ordenador con memoria ROM de 32K que contiene el sistema operativo QDOS, un sistema mono-usuario, multi-tarea y con partición de tiempo.

Para los que deseamos tener perfectamente ordenada nuestra agenda de trabajo, presupuestos, fichas de productos, nuestra correspondencia, estadísticas de venta, archivo... QL viene dotado de cuatro microdrives totalmente interactivados entre sí: QL QUILL de Tratamiento de Textos, QL ARCHIVE Base de Datos, QL ABACUS Hoja Electrónica de Cálculo y el QL EASEL para realización de todo tipo de gráficos.

Para los que nos gustan las cosas bien acabadas, QL se suministra con su fuente de alimentación, cables de conexión y adaptadores de TV, monitor y red local, cuatro programas de software de uso genérico, cuatro cartuchos en blanco para los microdrives y manual de instrucciones en castellano.

Para los que creemos que lo bien hecho puede tener también el mejor precio, QL el ordenador grande a precio pequeño.

Para los que nos gusta siempre ir bien acompañados, Sinclair —el mayor vendedor del mundo en ordenadores personales— e Investrónica, la mayor red de distribución de España, son nuestras mejores Compañías. Nuestra mejor garantía.

En definitiva, para los que queremos ordenarnos y nunca nos habíamos atrevido.

Con QL ya no hay excusas.



Tomás Bretón, 60. Telf. (91) 467 82 10. Télex 23399 IYCO E. 28045 Madrid
Camp, 80. Telf. (93) 211 26 58-211 27 54. 08022 Barcelona